



Guide de l'utilisateur

# AWS Schema Conversion Tool



Version 1.0.672

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Schema Conversion Tool: Guide de l'utilisateur

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Qu'est-ce que AWS SCT .....	1
Présentation de la conversion de schémas .....	5
Donner du feedback .....	7
Installation et configuration AWS SCT .....	8
Installation AWS SCT .....	8
Validation de l'installation .....	10
Vérification de la somme de contrôle du fichier AWS SCT .....	10
Vérifier les fichiers AWS SCT RPM sur Fedora .....	11
Vérifier les fichiers AWS SCT DEB sur Ubuntu .....	12
Vérification du fichier AWS SCT MSI sous Microsoft Windows .....	12
Installation des pilotes JDBC .....	13
Installation de pilotes JDBC sous Linux .....	17
Stockage des chemins des pilotes dans les paramètres globaux .....	18
Mise à jour AWS SCT .....	19
AWS Schema Conversion Tool CLI .....	20
AWS SCT interface utilisateur .....	21
Fenêtre du projet .....	21
Démarrage et gestion de projets .....	23
Utilisation de l'assistant .....	24
Sauvegarder des projets .....	28
Ajouter des serveurs de base de données .....	29
Mode hors ligne .....	30
Filtres arborescents .....	31
.....	32
Importation d'une liste de fichiers pour le filtre d'arborescence .....	34
Masquer les schémas .....	35
Rapport d'évaluation .....	37
Conversion de schémas .....	41
Appliquer des schémas convertis .....	44
Gestion des profils .....	45
Stockage des AWS identifiants .....	46
Définition du profil par défaut pour un projet .....	48
Autorisations d'utilisation du profil AWS de service .....	48
Configuration de Secrets Manager .....	49

Stockage des mots de passe de .....	50
Vue UNION ALL .....	50
Raccourcis clavier .....	51
Premiers pas .....	53
Connexion aux bases de données sources .....	55
Connexion à Amazon RDS et/ou Aurora chiffrés .....	56
Connexion à Apache Cassandra .....	59
Connexion à Apache Cassandra en tant que source .....	59
Connexion à Apache Hadoop .....	61
Conditions préalables à l'utilisation d'Apache Hadoop en tant que source .....	62
Autorisations pour Hive en tant que source .....	63
Autorisations pour HDFS en tant que source .....	63
Autorisations pour HDFS en tant que cible .....	64
Connexion à Apache Hadoop en tant que source .....	64
Connexion à Hive et HDFS .....	66
Connexion à Amazon EMR en tant que cible .....	69
Connexion à Apache Oozie .....	72
Prérequis .....	72
Connexion à Apache Oozie en tant que source .....	73
Autorisations pour AWS Lambda .....	75
Connexion à AWS Step Functions en tant que cible .....	77
Connexion à Azure SQL .....	78
Privilèges pour Azure SQL Database .....	78
Connexion à Azure SQL Database en tant que source .....	79
Connexion à IBM DB2 pour z/OS .....	81
Prérequis pour DB2 pour z/OS .....	81
Privilèges pour DB2 pour z/OS .....	82
Connexion à DB2 pour z/OS en tant que source .....	83
Privilèges pour MySQL en tant que cible .....	85
Privilèges pour PostgreSQL en tant que cible .....	87
Paramètres de conversion de DB2 pour z/OS vers PostgreSQL .....	88
Bases de données IBM DB2 LUW .....	90
Privilèges pour DB2 LUW .....	90
Connexion à DB2 LUW en tant que source .....	93
DB2 LUW vers PostgreSQL .....	95
DB2 LUW vers MySQL .....	98

Utilisation de MySQL comme source .....	99
Privilèges pour MySQL .....	100
Connexion à MySQL en tant que source .....	100
Privilèges pour PostgreSQL en tant que cible .....	103
Bases de données Oracle .....	104
Privilèges pour Oracle .....	105
Connexion à Oracle en tant que source .....	105
Oracle vers PostgreSQL .....	110
Oracle vers MySQL .....	116
Oracle vers Amazon RDS pour Oracle .....	126
Bases de données PostgreSQL .....	133
Privilèges pour PostgreSQL .....	134
Connexion à PostgreSQL en tant que source .....	134
Privilèges pour MySQL en tant que cible .....	137
Bases de données SAP .....	138
Privilèges pour SAP ASE .....	139
Connexion à SAP ASE en tant que source .....	140
Privilèges pour MySQL en tant que cible .....	142
Paramètres de conversion SAP ASE vers MySQL .....	144
Privilèges pour PostgreSQL en tant que cible .....	144
Paramètres de conversion de SAP ASE vers PostgreSQL .....	145
Bases de données SQL Server .....	146
Privilèges pour Microsoft SQL Server .....	147
Utilisation de l'authentification Windows avec Microsoft SQL Server .....	148
Connexion à SQL Server en tant que source .....	151
SQL Server vers MySQL .....	154
SQL Server vers PostgreSQL .....	159
SQL Server vers Amazon RDS SQL Server .....	195
Entrepôts de données .....	197
Amazon Redshift .....	197
Azure Synapse Analytics en tant que source .....	203
BigQuery comme source .....	209
Bases de données Greenplum .....	215
Bases de données Netezza .....	222
Entrepôt de données Oracle .....	232
Snowflake .....	240

Entrepôts de données SQL Server .....	249
Bases de données Teradata .....	256
Bases de données Vertica .....	273
Cartographie des types de données .....	280
Nouveau mappage des types de données .....	281
Modification des mappages de types de données .....	281
Cartographie de cibles virtuelles .....	283
Limites du mappage des types de données .....	284
Rapports .....	285
Rapport d'évaluation .....	285
Création d'un rapport d'évaluation .....	286
Afficher le rapport d'évaluation .....	287
Sauvegarde du rapport d'évaluation .....	291
Configuration d'un rapport d'évaluation .....	293
Rapport d'évaluation multiserveur .....	297
Conversion de schémas .....	307
Appliquer les règles de migration .....	309
Création de règles de migration .....	310
Exportation des règles de migration .....	312
Conversion manuelle de schémas .....	312
Conversion de schémas .....	313
Modification du schéma converti .....	316
Effacement d'un schéma converti .....	317
Conversion manuelle de schémas .....	317
Modification de votre schéma source .....	317
Modification de votre schéma cible .....	317
Mise à jour et actualisation des schémas .....	318
Enregistrement et application de schémas convertis .....	319
Sauvegarde de votre schéma converti .....	319
Appliquer votre schéma converti .....	320
Le schéma du pack d'extension .....	321
Comparaison de schémas .....	321
Affichage des objets transformés associés .....	323
Conversion de schémas d'entrepôts de données .....	324
Autorisations pour Amazon Redshift .....	326
Choix des stratégies et des règles d'optimisation .....	327

Collecte ou téléchargement de statistiques .....	329
Création de règles de migration .....	330
Création de règles de migration .....	331
Exportation des règles de migration .....	333
Conversion de votre schéma .....	333
Conversion de schémas .....	334
Modification du schéma converti .....	336
Effacement d'un schéma converti .....	337
Gestion et personnalisation des clés .....	337
Rubriques en relation .....	338
Création et utilisation du rapport d'évaluation .....	338
Création d'un rapport d'évaluation de migration de base de données .....	339
Récapitulatif .....	339
Éléments d'action .....	341
Sauvegarde du rapport d'évaluation .....	341
Gestion des conversions manuelles .....	342
Modification de votre schéma source .....	343
Modification de votre schéma cible .....	343
Mise à jour et actualisation de votre schéma converti .....	344
Enregistrer et appliquer votre schéma converti .....	344
Enregistrement de votre schéma converti dans un fichier .....	345
Appliquer votre schéma converti .....	346
Le schéma du pack d'extension .....	346
Bibliothèques Python .....	347
Conversion de données Amazon Redshift .....	347
Optimisation de votre base de données Amazon Redshift .....	347
Conversion de données à l'aide d'ETL .....	350
Processus ETL .....	351
Prérequis .....	352
AWS Glue Catalogue de données .....	353
Limites .....	353
Étape 1 : Créer un projet .....	355
Étape 2 : Création d'un AWS Glue emploi .....	356
Processus ETL utilisant Python .....	358
Étape 1 : Créer une base de données .....	358
Étape 2 : Création d'une connexion .....	359

Étape 3 : créer un AWS Glue crawler .....	360
Scripts ETL Informatica .....	363
Paquets SSIS .....	368
Composants SSIS pris en charge .....	372
SSIS vers AWS Glue Studio .....	374
Prérequis .....	374
Ajouter des packages SSIS à votre AWS SCT projet .....	376
Conversion de packages SSIS .....	377
Création d' AWS Glue Studio emplois .....	378
Création d'un rapport d'évaluation des conversions SSIS .....	380
Composants SSIS pris en charge .....	381
Scripts BTEQ Teradata .....	382
Ajouter des scripts BTEQ à votre projet AWS SCT .....	384
Configuration des variables de substitution dans les scripts BTEQ .....	385
Conversion de scripts BTEQ .....	386
Gestion des scripts BTEQ .....	386
Création d'un rapport d'évaluation de la conversion des scripts BTEQ .....	387
Modification et enregistrement de vos scripts BTEQ convertis .....	388
Scripts shell .....	388
Ajouter des scripts shell à votre AWS SCT projet .....	389
Configuration des variables de substitution dans les scripts shell .....	390
Conversion de scripts shell .....	391
Gestion des scripts shell .....	392
Création d'un rapport d'évaluation de la conversion d'un script shell .....	392
Modification et enregistrement de vos scripts shell convertis .....	393
FastExport scripts .....	394
Ajouter des scripts de FastExport travail à votre AWS SCT projet .....	394
Configuration des variables de substitution dans les scripts de FastExport travail .....	395
Conversion de scripts de FastExport tâches .....	397
Gestion des scripts de FastExport tâches .....	398
Création d'un rapport d'évaluation de la conversion des scripts de FastExport travail .....	398
Modification et enregistrement de vos scripts de FastExport travail convertis .....	399
FastLoad scripts .....	400
Ajouter des scripts de FastLoad travail à votre AWS SCT projet .....	400
Configuration des variables de substitution dans les scripts de FastLoad travail .....	402
Conversion de scripts de FastLoad tâches .....	403



Gestion des scripts de FastLoad tâches .....	404
Création d'un rapport d'évaluation de la conversion des scripts de FastLoad travail .....	405
Modification et enregistrement de vos scripts de FastLoad travail convertis .....	406
MultiLoad scripts .....	406
Ajouter des scripts de MultiLoad travail à votre AWS SCT projet .....	407
Configuration des variables de substitution dans les scripts de MultiLoad travail .....	408
Conversion de scripts de MultiLoad tâches .....	409
Gestion des scripts de MultiLoad tâches .....	410
Création d'un rapport d'évaluation de la conversion des scripts de MultiLoad travail .....	411
Modification et enregistrement de vos scripts de MultiLoad travail convertis .....	412
Migration des infrastructures de mégadonnées .....	413
Migration des charges de travail Hadoop .....	413
Présentation .....	414
Étape 1 : Connectez-vous à vos clusters Hadoop .....	415
Étape 2 : configurer les règles de mappage .....	415
Étape 3 : Création d'un rapport d'évaluation .....	417
Étape 4 : migrer votre cluster Apache Hadoop vers Amazon EMR .....	418
Exécution de votre script CLI .....	419
Gestion de votre projet de migration .....	420
Conversion des flux de travail Oozie ; .....	422
Présentation .....	422
Étape 1 : Connectez-vous à vos services source et cible .....	424
Étape 2 : configurer les règles de mappage .....	424
Étape 3 : Configuration des paramètres .....	425
Étape 4 : Créer un rapport d'évaluation .....	427
Étape 5 : Convertissez vos flux de travail Apache Oozie en AWS Step Functions .....	428
Exécution de votre script CLI .....	431
Nœuds supportés .....	431
Intégration avec AWS DMS .....	433
Utilisation d'un agent AWS SCT de réplication avec AWS DMS .....	433
Utilisation d'un agent d'extraction de AWS SCT données avec AWS DMS .....	433
Augmenter les niveaux de journalisation lors de l'utilisation AWS SCT avec AWS DMS .....	434
Migration depuis un entrepôt de données .....	436
Prérequis .....	439
Paramètres Amazon S3 .....	439
Assumer des rôles IAM .....	440

Paramètres de sécurité .....	442
Paramètres de configuration .....	443
Installation des agents .....	443
Configuration des agents .....	445
Installation et configuration d'agents de copie dédiés .....	447
Agents de départ .....	449
Enregistrement des agents .....	450
Masquer et récupérer des informations pour un AWS SCT agent .....	450
Création de règles de migration des données .....	452
Modification des paramètres d'extracteur et de copie pour la migration des données .....	454
Tri des données .....	457
Création, exécution et surveillance d'une AWS SCT tâche .....	459
Exportation et importation d'une tâche d'extraction de données .....	462
Extraction de données à l'aide d'un appareil AWS Snowball Edge Edge .....	463
Step-by-step procédures de migration des données à l'aide AWS SCT d'Edge AWS Snowball Edge .....	464
Sortie de la tâche d'extraction de données .....	468
Utilisation du partitionnement virtuel .....	469
Limites lors de la création d'un partitionnement virtuel .....	470
Type de cloison RANGE .....	470
Type de partition LIST .....	471
Type de partition DATE AUTO SPLIT .....	473
Utilisation du partitionnement natif .....	474
Travailler avec LOBs .....	475
Bonnes pratiques et résolution des problèmes .....	476
Conversion du code SQL d'une application .....	478
Présentation de la conversion du code SQL d'une application .....	478
Code SQL .....	479
Création de projets de conversion d'applications génériques .....	479
Gestion des projets de conversion d'applications .....	484
Analyse et conversion de votre code SQL .....	485
Création et utilisation du rapport d'évaluation .....	486
Modification et enregistrement de votre code SQL converti .....	487
Code SQL dans les applications C# .....	488
Création de projets de conversion d'applications C# .....	488
Conversion du code SQL de votre application C# .....	489

Sauvegarde du code d'application converti .....	491
Gestion des projets de conversion d'applications C# .....	491
Création d'un rapport d'évaluation de la conversion des applications C# .....	492
Code SQL en C++ .....	493
Création de projets de conversion d'applications C++ .....	494
Conversion du code SQL de votre application C++ .....	495
Sauvegarde du code d'application converti .....	497
Gestion des projets de conversion d'applications C++ .....	498
Création d'un rapport d'évaluation de la conversion des applications C++ .....	499
Code SQL en Java .....	500
Création de projets de conversion d'applications Java .....	501
Conversion du code SQL de votre application Java .....	503
Sauvegarde du code d'application converti .....	505
Gestion des projets de conversion d'applications Java .....	505
Création d'un rapport d'évaluation de la conversion d'une application Java .....	506
Code SQL dans Pro*C .....	508
Création de projets de conversion d'applications Pro*C .....	508
Conversion du code SQL de votre application Pro*C .....	510
Modification et enregistrement du code d'application converti .....	511
Gestion des projets de conversion d'applications Pro*C .....	512
Création d'un rapport d'évaluation de la conversion d'une application Pro*C .....	513
Packs d'extension .....	515
Autorisations d'utilisation du pack d'extension .....	517
Utilisation du schéma du pack d'extension .....	518
Bibliothèques personnalisées pour les packs d'extension .....	518
Appliquer le pack d'extension .....	519
Utilisation des fonctions Lambda du pack d'extension AWS SCT .....	521
Utilisation de AWS Lambda fonctions pour émuler les fonctionnalités d'une base de données .....	522
Appliquer le pack d'extension pour prendre en charge les fonctions Lambda .....	522
Configuration des fonctions du pack d'extension .....	524
Bonnes pratiques .....	526
Configuration de mémoire supplémentaire .....	526
Dossier de projet par défaut .....	526
Augmenter la vitesse de migration des données .....	527
Augmenter les informations de journalisation .....	527

Résolution des problèmes .....	531
Je ne peux pas charger d'objets à partir d'une base de données source Oracle .....	531
Message d'avertissement .....	531
CLI Reference .....	533
Prérequis .....	533
Mode interactif .....	533
Exemples .....	535
Obtenir des scénarios CLI .....	535
Exemples .....	540
Modification de scénarios CLI .....	540
Mode script .....	542
Exemples .....	543
Matériel de référence .....	543
Notes de mise à jour .....	544
Notes de mise à jour — 676 .....	544
Notes de mise à jour — 675 .....	549
Notes de mise à jour — 674 .....	552
Notes de mise à jour — 673 .....	559
Notes de mise à jour — 672 .....	564
Notes de mise à jour — 671 .....	573
Notes de mise à jour — 670 .....	582
Notes de mise à jour — 669 .....	587
Notes de mise à jour — 668 .....	593
Notes de mise à jour — 667 .....	599
Notes de mise à jour — 666 .....	603
Notes de mise à jour — 665 .....	608
Notes de mise à jour — 664 .....	611
Notes de mise à jour — 663 .....	615
Notes de mise à jour — 662 .....	618
Notes de mise à jour — 661 .....	623
Notes de mise à jour — 660 .....	628
Notes de mise à jour — 659 .....	632
Notes de mise à jour — 658 .....	636
Notes de mise à jour — 657 .....	641
Notes de mise à jour — 656 .....	645
Notes de mise à jour — 655 .....	649

Notes de mise à jour — 654 .....	652
Notes de mise à jour — 653 .....	655
Notes de mise à jour — 652 .....	657
Notes de mise à jour — 651 .....	659
Notes de mise à jour — 650 .....	662
Notes de mise à jour — 649 .....	664
Notes de mise à jour — 648 .....	667
Notes de mise à jour — 647 .....	668
Notes de mise à jour — 646 .....	670
Notes de mise à jour — 645 .....	672
Notes de mise à jour — 644 .....	673
Notes de mise à jour — 642 .....	675
Notes de mise à jour — 641 .....	677
Notes de mise à jour — 640 .....	678
Modifications apportées à Oracle dans la version 1.0.640 .....	678
Modifications apportées à la version 1.0.640 de Microsoft SQL Server .....	685
Modifications de la version 1.0.640 pour MySQL .....	689
Modifications apportées à PostgreSQL dans la version 1.0.640 .....	690
Modifications apportées à la version 1.0.640 de DB2 LUW .....	693
Modifications apportées à Teradata dans la version 1.0.640 .....	694
Modifications apportées à la version 1.0.640 pour les autres moteurs .....	696
Historique de la documentation .....	699
Mises à jour antérieures .....	717
.....	dccxxvii

# Qu'est-ce que c'est AWS Schema Conversion Tool ?

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir votre schéma de base de données existant d'un moteur de base de données à un autre. Vous pouvez convertir le schéma OLTP relationnel ou le schéma d'entrepôt de données. Votre schéma converti convient à une base de données Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, PostgreSQL, à un cluster de base de données Amazon Aurora ou à un cluster Amazon Redshift. Le schéma converti peut également être utilisé avec une base de données sur une EC2 instance Amazon ou stocké sous forme de données dans un compartiment Amazon S3.

AWS SCT prend en charge plusieurs normes du secteur, notamment les normes fédérales de traitement de l'information (FIPS), pour les connexions à un compartiment Amazon S3 ou à une autre AWS ressource. AWS SCT est également conforme au programme fédéral de gestion des risques et des autorisations (FedRAMP). Pour plus de détails sur AWS les efforts de conformité et les efforts de mise [en conformité, voir les AWS services concernés par le programme de conformité](#).

AWS SCT prend en charge les conversions OLTP suivantes.

Base de données source	Base de données cible
IBM DB2 pour z/OS (version 12)	Édition compatible avec Amazon Aurora MySQL (Aurora MySQL), édition compatible avec Amazon Aurora PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL  Pour de plus amples informations, veuillez consulter <a href="#">Connexion à IBM DB2 pour z/OS</a> .
IBM Db2 LUW (versions 9.1, 9.5, 9.7, 10.5, 11.1 et 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL  Pour de plus amples informations, veuillez consulter <a href="#">Bases de données IBM DB2 LUW</a> .
Base de données Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Base de données source	Base de données cible
	<p>Pour de plus amples informations, veuillez consulter <a href="#">Connexion à Azure SQL</a>.</p>
Microsoft SQL Server (versions 2008 R2, 2012, 2014, 2016, 2017, 2019 et 2022)	<p>Aurora MySQL, Aurora PostgreSQL, Babelfish pour Aurora PostgreSQL (uniquement pour les rapports d'évaluation), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Bases de données SQL Server</a>.</p>
MySQL (version 5.5 et supérieure)	<p>Aurora PostgreSQL, MySQL, PostgreSQL</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Utilisation de MySQL comme source</a>.</p> <p>Vous pouvez migrer le schéma et les données de MySQL vers un cluster de base de données Aurora MySQL sans utiliser AWS SCT. Pour plus d'informations, consultez la section <a href="#">Migration des données vers un cluster de base de données Amazon Aurora</a>.</p>
Oracle (version 10.1 et supérieure)	<p>Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Bases de données Oracle</a>.</p>
PostgreSQL (version 9.1 et supérieure)	<p>Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Bases de données PostgreSQL</a>.</p>

Base de données source	Base de données cible
SAP ASE (versions 12.5.4, 15.0.2, 15.5, 15.7 et 16.0)	<p>Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Bases de données SAP</a>.</p>

AWS SCT prend en charge les conversions d'entrepôts de données suivantes.

Entrepôt de données source	Entrepôt de données cible
Amazon Redshift	<p>Amazon Redshift</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Amazon Redshift</a>.</p>
Analyses Azure Synapse	<p>Amazon Redshift</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Azure Synapse Analytics en tant que source</a>.</p>
BigQuery	<p>Amazon Redshift</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">BigQuery comme source</a>.</p>
Base de données Greenplum (versions 4.3 et 6.21)	<p>Amazon Redshift</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Bases de données Greenplum</a>.</p>
Microsoft SQL Server (version 2008 et supérieure)	<p>Amazon Redshift</p> <p>Pour de plus amples informations, veuillez consulter <a href="#">Entrepôts de données SQL Server</a>.</p>
Netezza (version 7.0.3 et supérieure)	Amazon Redshift



Entrepôt de données source	Entrepôt de données cible
	Pour de plus amples informations, veuillez consulter <a href="#">Bases de données Netezza</a> .
Oracle (version 10.1 et supérieure)	Amazon Redshift  Pour de plus amples informations, veuillez consulter <a href="#">Entrepôt de données Oracle</a> .
Flocon de neige (version 3)	Amazon Redshift  Pour de plus amples informations, veuillez consulter <a href="#">Snowflake</a> .
Teradata (version 13 et supérieure)	Amazon Redshift  Pour de plus amples informations, veuillez consulter <a href="#">Bases de données Teradata</a> .
Vertica (version 7.2.2 et supérieure)	Amazon Redshift  Pour de plus amples informations, veuillez consulter <a href="#">Bases de données Vertica</a> .

AWS SCT prend en charge les conversions de base de données NoSQL suivantes.

Base de données source	Base de données cible
Apache Cassandra (versions 2.1.x, 2.2.16 et 3.11.x)	Amazon DynamoDB  Pour de plus amples informations, veuillez consulter <a href="#">Connexion à Apache Cassandra</a> .

AWS SCT prend en charge les conversions des processus d'extraction, de transformation et de chargement (ETL) suivants. Pour de plus amples informations, veuillez consulter [Conversion de données à l'aide d'ETL](#).

Source	Cible
Scripts ETL Informatica	Informatica
Packages ETL Microsoft SQL Server Integration Services (SSIS)	AWS Glue ou AWS Glue Studio
Scripts Shell avec commandes intégrées à partir de Teradata Basic Teradata Query (BTEQ)	Amazon Redshift RSQL
Scripts ETL Teradata BTEQ	AWS Glue ou Amazon Redshift RSQL
Scripts de FastExport tâches Teradata	Amazon Redshift RSQL
Scripts de FastLoad tâches Teradata	Amazon Redshift RSQL
Scripts de MultiLoad tâches Teradata	Amazon Redshift RSQL

AWS SCT prend en charge les migrations de framework Big Data suivantes. Pour de plus amples informations, veuillez consulter [Migration des infrastructures de mégadonnées](#).

Source	Cible
Apache Hive (version 0.13.0 et supérieure)	Hive sur Amazon EMR
Apache HDFS	Amazon S3 ou HDFS sur Amazon EMR
Apache Oozie	AWS Step Functions

## Présentation de la conversion de schémas

AWS SCT fournit une interface utilisateur basée sur un projet pour convertir automatiquement le schéma de base de données de votre base de données source dans un format compatible avec votre instance Amazon RDS cible. Si le schéma de votre base de données source ne peut pas être converti automatiquement, AWS SCT fournit des conseils sur la manière de créer un schéma équivalent dans votre base de données Amazon RDS cible.

Pour plus d'informations sur la procédure d'installation AWS SCT, consultez [Installation et configuration AWS Schema Conversion Tool](#).

Pour une présentation de l'interface AWS SCT utilisateur, voir [Naviguer dans l'interface utilisateur du AWS SCT](#).

Pour plus d'informations sur le processus de conversion, consultez [Conversion de schémas de base de données dans AWS Schema Conversion Tool](#).

Outre la conversion de votre schéma de base de données existant d'un moteur de base de données à un autre, AWS SCT certaines fonctionnalités supplémentaires vous aident à déplacer vos données et applications vers le AWS cloud :

- Vous pouvez utiliser des agents d'extraction de données pour extraire les données de votre entrepôt de données afin de préparer leur migration vers Amazon Redshift. Pour gérer les agents d'extraction de données, vous pouvez utiliser AWS SCT. Pour de plus amples informations, veuillez consulter [Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool](#).
- Vous pouvez l'utiliser AWS SCT pour créer des AWS DMS points de terminaison et des tâches. Vous pouvez exécuter et surveiller ces tâches depuis AWS SCT. Pour de plus amples informations, veuillez consulter [Intégration AWS Database Migration Service avec AWS Schema Conversion Tool](#).
- Dans certains cas, les fonctionnalités de base de données ne peuvent pas être converties en fonctionnalités Amazon RDS ou Amazon Redshift équivalentes. L'assistant du pack d' AWS SCT extension peut vous aider à installer AWS Lambda des fonctions et des bibliothèques Python pour émuler les fonctionnalités qui ne peuvent pas être converties. Pour de plus amples informations, veuillez consulter [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).
- Vous pouvez l'utiliser AWS SCT pour optimiser votre base de données Amazon Redshift existante. AWS SCT recommande des clés de tri et des clés de distribution pour optimiser votre base de données. Pour de plus amples informations, veuillez consulter [Conversion de données depuis Amazon Redshift à l'aide de AWS Schema Conversion Tool](#).
- Vous pouvez l'utiliser AWS SCT pour copier votre schéma de base de données sur site existant vers une instance de base de données Amazon RDS exécutant le même moteur. Cette fonction vous permet d'analyser les économies de coûts potentielles en cas de déplacement vers le cloud et de changement de votre type de licence.
- Vous pouvez utiliser AWS SCT pour convertir le code SQL dans le code de votre application C++, C#, Java ou autre. Vous pouvez afficher, analyser, modifier et enregistrer le code SQL converti.

Pour de plus amples informations, veuillez consulter [Conversion du code SQL d'une application en utilisant AWS SCT](#).

- Vous pouvez l'utiliser AWS SCT pour migrer les processus d'extraction, de transformation et de chargement (ETL). Pour de plus amples informations, veuillez consulter [Conversion de données à l'aide de processus ETL dans AWS Schema Conversion Tool](#).

## Fournir des commentaires

Vous pouvez fournir des commentaires sur AWS SCT. Vous pouvez déposer un rapport de bogue, soumettre une demande de fonctionnalité ou fournir des informations générales.

Pour fournir des commentaires sur AWS SCT

1. Démarrez le AWS Schema Conversion Tool.
2. Ouvrez le menu Help, puis choisissez Leave Feedback. La boîte de dialogue Leave Feedback s'affiche.
3. Dans Area, choisissez Information, Bug report ou Feature request.
4. Dans Source database, choisissez votre base de données source. Choisissez Any si vos commentaires ne concernent pas une base de données spécifique.
5. Dans Target database, choisissez votre base de données cible. Choisissez Any si vos commentaires ne concernent pas une base de données spécifique.
6. Dans Title, tapez un titre pour vos commentaires.
7. Dans Message, tapez vos commentaires.
8. Choisissez Send pour soumettre vos commentaires.

# Installation et configuration AWS Schema Conversion Tool

The AWS Schema Conversion Tool (AWS SCT) est une application autonome qui fournit une interface utilisateur basée sur des projets. AWS SCT est disponible pour Microsoft Windows, Fedora Linux et Ubuntu Linux. AWS SCT n'est pris en charge que sur les systèmes d'exploitation 64 bits.

Pour garantir que vous obtenez la bonne version du fichier de AWS SCT distribution, nous vous indiquons les étapes de vérification après le téléchargement du fichier compressé. Vous pouvez alors vérifier le fichier à l'aide des étapes fournies.

AWS SCT est disponible à la fois en tant qu'application autonome et en tant qu'outil de ligne de commande. Pour plus d'informations sur l'outil de ligne de commande, consultez [AWS Schema Conversion Tool CLI](#).

## Rubriques

- [Installation AWS Schema Conversion Tool](#)
- [Validation de l'installation AWS Schema Conversion Tool](#)
- [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#)
- [Mise à jour AWS Schema Conversion Tool](#)
- [AWS Schema Conversion Tool CLI](#)

## Installation AWS Schema Conversion Tool

Vous pouvez effectuer l'installation AWS SCT sur les systèmes d'exploitation suivants :

- Microsoft Windows 10
- Fedora Linux 36 et versions ultérieures
- Ubuntu Linux 18 et versions ultérieures

### Pour installer AWS SCT

1. Téléchargez le fichier compressé contenant le AWS SCT programme d'installation à l'aide du lien correspondant à votre système d'exploitation. Tous les fichiers compressés ont une extension .zip. Lorsque vous extrayez le fichier AWS SCT d'installation, celui-ci sera au format adapté à votre système d'exploitation.

- [Microsoft Windows](#)
- [Ubuntu Linux \(.deb\)](#)
- [Fedora Linux \(.rpm\)](#)

2. Extrayez le fichier d' AWS SCT installation correspondant à votre système d'exploitation, comme indiqué ci-dessous.

Système d'exploitation	Nom de fichier
Fedora Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm
Microsoft Windows	AWS Schema Conversion Tool-1.0. <i>build-number</i> .msi
Ubuntu Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .deb

3. Exécutez le fichier AWS SCT d'installation extrait à l'étape précédente. Suivez les instructions ci-dessous correspondant à votre système d'exploitation.

Système d'exploitation	Instructions d'installation
Fedora Linux	<p>Exécutez la commande suivante dans le dossier où vous avez téléchargé le fichier :</p> <pre>sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm</pre>
Microsoft Windows	Double-cliquez sur le fichier pour exécuter le programme d'installation.
Ubuntu Linux	<p>Exécutez la commande suivante dans le dossier où vous avez téléchargé le fichier :</p> <pre>sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i> .deb</pre>

4. Téléchargez les pilotes Java Database Connectivity (JDBC) pour vos moteurs de base de données source et cible. Vous trouverez des instructions et des liens de téléchargement à la page [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

Vous avez maintenant terminé la configuration de l' AWS SCT application. Double-cliquez sur l'icône de l'application pour l'exécuter AWS SCT.

## Validation de l'installation AWS Schema Conversion Tool

Vous pouvez vérifier le fichier de distribution de plusieurs manières AWS SCT. Le plus simple est de comparer le checksum du fichier avec le checksum publié à partir de. AWS Pour encore plus de sécurité, vous pouvez utiliser les procédures ci-dessous afin de vérifier le fichier de distribution en fonction du système d'exploitation sur lequel vous avez installé le fichier.

Cette section comprend les rubriques suivantes.

### Rubriques

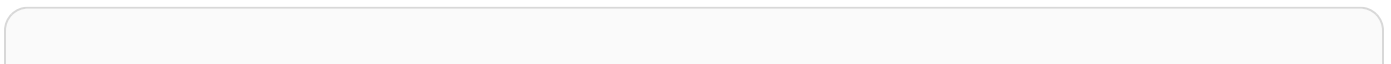
- [Vérification de la somme de contrôle du fichier AWS SCT](#)
- [Vérifier les fichiers AWS SCT RPM sur Fedora](#)
- [Vérifier les fichiers AWS SCT DEB sur Ubuntu](#)
- [Vérification du fichier AWS SCT MSI sous Microsoft Windows](#)

## Vérification de la somme de contrôle du fichier AWS SCT

Afin de détecter les erreurs qui auraient pu être introduites lors du téléchargement ou du stockage du fichier AWS SCT compressé, vous pouvez comparer la somme de contrôle du fichier à une valeur fournie par AWS. AWS utilise l' SHA256 algorithme pour la somme de contrôle.

Pour vérifier le fichier de AWS SCT distribution à l'aide d'une somme de contrôle

1. Téléchargez le fichier AWS SCT de distribution à l'aide des liens de la section Installation. Pour de plus amples informations, veuillez consulter [Installation AWS Schema Conversion Tool](#).
2. Téléchargez le dernier fichier de total de contrôle, appelé [sha256Check.txt](#). Ce fichier inclut les sommes de contrôle de la dernière AWS SCT version. Par exemple, le fichier peut apparaître comme suit :



```
Fedora    b4f5f66f91bfcc1b312e2827e960691c269a9002cd1371cf1841593f88cbb5e6
Ubuntu    4315eb666449d4fcd95932351f00399adb6c6cf64b9f30adda2eec903c54eca4
Windows   6e29679a3c53c5396a06d8d50f308981e4ec34bd0acd608874470700a0ae9a23
```

3. Exécutez la commande de SHA256 validation pour votre système d'exploitation dans le répertoire contenant le fichier de distribution. Par exemple, exécutez la commande suivante sous Linux.

```
shasum -a 256 aws-schema-conversion-tool-1.0.latest.zip
```

4. Comparez les résultats de la commande avec la valeur affichée dans le fichier sha256Check.txt. Si les sommes de contrôle correspondent, vous pouvez exécuter le fichier de distribution en toute sécurité. Si les sommes de contrôle ne correspondent pas, n'exécutez pas le fichier de distribution et [contactez AWS Support](#).

## Vérifier les fichiers AWS SCT RPM sur Fedora

AWS fournit un autre niveau de validation en plus de la somme de contrôle du fichier de distribution. Tous les fichiers RPM du fichier de distribution sont signés par une clé AWS privée. La clé GPG publique peut être consultée à l'adresse [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).

Pour vérifier les fichiers AWS SCT RPM sur Fedora

1. Téléchargez le fichier AWS SCT de distribution à l'aide des liens de la section Installation.
2. Vérifiez la somme de contrôle du fichier de AWS SCT distribution.
3. Extrayez le contenu du fichier de distribution. Recherchez le fichier RPM que vous souhaitez vérifier.
4. Téléchargez la clé GPG publique à partir de [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key)
5. Importez la clé publique dans votre base de données RPM (assurez-vous que vous avez les autorisations appropriées) à l'aide de la commande suivante :

```
sudo rpm --import aws-dms-team@amazon.com.public.gpg-key
```

6. Vérifiez que l'importation a réussi en exécutant la commande suivante :



```
rpm -q --qf "%{NAME}-%{VERSION}-%{RELEASE} \n %{SUMMARY} \n" gpg-pubkey-  
ea22abf4-5a21d30c
```

7. Vérifiez la signature RPM en exécutant la commande suivante :

```
rpm --checksig -v aws-schema-conversion-tool-1.0.build number-1.x86_64.rpm
```

## Vérifier les fichiers AWS SCT DEB sur Ubuntu

AWS fournit un autre niveau de validation en plus de la somme de contrôle du fichier de distribution. Tous les fichiers DEB du fichier de distribution sont signés par une signature détachée GPG.

Pour vérifier les fichiers AWS SCT DEB sur Ubuntu

1. Téléchargez le fichier AWS SCT de distribution à l'aide des liens de la section Installation.
2. Vérification de la somme de contrôle du fichier de AWS SCT distribution.
3. Extrayez le contenu du fichier de distribution. Recherchez le fichier DEB que vous souhaitez vérifier.
4. Téléchargez la signature détachée depuis [aws-schema-conversion-tool-1.0.latest.deb.asc](https://aws-schema-conversion-tool-1.0.latest.deb.asc).
5. Téléchargez la clé GPG publique à partir de [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).
6. Importez la clé publique GPG en exécutant la commande suivante :

```
gpg --import aws-dms-team@amazon.com.public.gpg-key
```

7. Vérifiez la signature en exécutant la commande suivante :

```
gpg --verify aws-schema-conversion-tool-1.0.latest.deb.asc aws-schema-conversion-  
tool-1.0.build number.deb
```

## Vérification du fichier AWS SCT MSI sous Microsoft Windows

AWS fournit un autre niveau de validation en plus de la somme de contrôle du fichier de distribution. Le fichier MSI possède une signature numérique que vous pouvez vérifier pour vous assurer qu'il a bien été signé AWS.

## Pour vérifier le fichier AWS SCT MSI sous Windows

1. Téléchargez le fichier AWS SCT de distribution à l'aide des liens de la section Installation.
2. Vérification de la somme de contrôle du fichier de AWS SCT distribution.
3. Extrayez le contenu du fichier de distribution. Recherchez le fichier MSI que vous souhaitez vérifier.
4. Dans l'explorateur Windows, cliquez avec le bouton droit de la souris sur le fichier MSI et sélectionnez Propriétés.
5. Choisissez l'onglet Signatures numériques.
6. Vérifiez que la signature numérique provient d'Amazon Services LLC.

## Installation des pilotes JDBC pour AWS Schema Conversion Tool

AWS SCT Pour fonctionner correctement, téléchargez les pilotes JDBC pour vos moteurs de base de données source et cible. Si vous utilisez une plate-forme de base de données cible virtuelle, vous n'avez pas besoin de télécharger le pilote JDBC pour votre moteur de base de données cible. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

Après avoir téléchargé les pilotes, indiquez l'emplacement des fichiers de pilote. Pour de plus amples informations, veuillez consulter [Stockage des chemins des pilotes dans les paramètres globaux](#).


Vous pouvez télécharger les pilotes de base de données à partir des emplacements suivants.

### Important

Téléchargez la dernière version du pilote disponible. Le tableau suivant indique la version la plus basse du pilote de base de données prise en charge par AWS SCT.

Moteur de base de données	Pilotes	Emplacement de téléchargement
Amazon Aurora	mysql-connector-java-5.1.6.jar	<a href="https://www.mysql.com/products/connecteur/">https://www.mysql.com/products/connecteur/</a>

Moteur de base de données	Pilotes	Emplacement de téléchargement
MySQL- Compatible Edition		
Amazon Aurora PostgreSQL- Compatible Edition	postgresql-42.2.19.jar	<a href="https://jdbc.postgresql.org/download/postgresql-42.2.19.jar">https://jdbc.postgresql.org/download/postgresql-42.2.19.jar</a>
Amazon EMR	HiveJDBC42.jar	<a href="http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/">http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/</a>
Amazon Redshift	redshift-jdbc42-2.1.0.9.jar	<a href="https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip">https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip</a>
Amazon Redshift sans serveur	redshift-jdbc42-2.1.0.9.jar	<a href="https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip">https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip</a>
Apache Hive	hive-jdbc-2.3.4-standalone.jar	<a href="https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar">https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar</a>
Base de données Azure SQL	mssql-jdbc-7.2.2.jre11.jar	<a href="https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver-15#72">https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver-15#72</a>
Analyses Azure Synapse	mssql-jdbc-7.2.2.jre11.jar	<a href="https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver-15#72">https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver-15#72</a>
Greenplum Database	postgresql-42.2.19.jar	<a href="https://jdbc.postgresql.org/download/postgresql-42.2.19.jar">https://jdbc.postgresql.org/download/postgresql-42.2.19.jar</a>

Moteur de base de données	Pilotes	Emplacement de téléchargement
IBM DB2 pour z/OS	db2jcc-db2jcc4.jar	<a href="https://www.ibm.com/support/pages/db2--downloads-db2-zos-jdbc-driver-versions-and">https://www.ibm.com/support/pages/db2- - downloads-db2-zos jdbc-driver-versions-and</a>
IBM Db2 LUW	db2jcc-db2jcc4.jar	<a href="https://www.ibm.com/support/pages/node/382667">https://www.ibm.com/support/pages/node/382667</a>
MariaDB	mariadb-java-client-2.4.1.jar	<a href="https://downloads.mariadb.com/Connectors/java/connector-java-2.4.1/mariadb-java-client-2.4.1.jar">https://downloads.mariadb.com/Connectors/java/connector-java-2.4.1/mariadb-java-client-2.4.1.jar</a>
Microsoft SQL Server	mssql-jdbc-10.2.jar	<a href="https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?vue=15sql-server-ver">https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?vue=15sql-server-ver</a>
		<p> <b>Note</b></p> <p>AWS SCT ne prend pas en charge la dernière version du pilote JDBC 18.2.1.0 pour MSSQL. Nous vous recommandons d'installer la version 10.2 du pilote JDBC mssql-jdbc-10.2.jar).</p>
MySQL	mysql-connector-java-8.0.15.jar	<a href="https://dev.mysql.com/downloads/connector/j/">https://dev.mysql.com/downloads/connector/j/</a>
Netezza	nzjdbc.jar	<a href="http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html">http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html</a>
	Utiliser les logiciels d'outils clients. Téléchargez la version 7.2.1 du pilote, qui est rétrocompatible avec la version 7.2.0 de l'entrepôt de données.	

Moteur de base de données	Pilotes	Emplacement de téléchargement
Oracle	ojdbc8.jar  Les versions 8 et supérieures du pilote sont prises en charge.	<a href="https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html">https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html</a>
PostgreSQL	postgresql-42.2.19.jar	<a href="https://jdbc.postgresql.org/download/postgresql-42.2.19.jar">https://jdbc.postgresql.org/download/postgresql-42.2.19.jar</a>
SAP ASE (Sybase ASE)	jconn4.jar	<a href="#">Le pilote JDBC jConnect</a>
Snowflake	snowflake-jdbc-3.9.2.jar  Pour plus d'informations, voir <a href="#">Télécharger/intégrer le pilote JDBC.</a>	<a href="https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar">https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar</a>
Teradata	terajdbc4.jar  tdgssconfig.jar  Pour les versions 16.20.00.11 et supérieures du pilote Teradata JDBC, vous n'avez pas besoin du fichier. tdgssconfig.jar	<a href="https://downloads.teradata.com/download/connectivité/pilote JDBC">https://downloads.teradata.com/download/connectivité/pilote JDBC</a>

Moteur de base de données	Pilotes	Emplacement de téléchargement
Vertica	vertica-jdbc-9.1.1-0.jar  Les versions 7.2.0 et supérieures du pilote sont prises en charge.	<a href="https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar">https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar</a>

## Installation de pilotes JDBC sous Linux

Vous pouvez suivre les étapes ci-dessous pour installer les pilotes JDBC sur votre système Linux afin de les utiliser avec AWS SCT.

Pour installer les pilotes JDBC sur votre système Linux

1. Créez un répertoire pour y stocker les pilotes JDBC.

```
PROMPT>sudo mkdir -p /usr/local/jdbc-drivers
```

2. Installez le pilote JDBC pour votre moteur de base de données à l'aide des commandes indiquées ci-dessous.

Moteur de base de données	Commandes d'installation
Amazon Aurora (compatible MySQL)	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Amazon Aurora (compatible PostgreSQL)	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

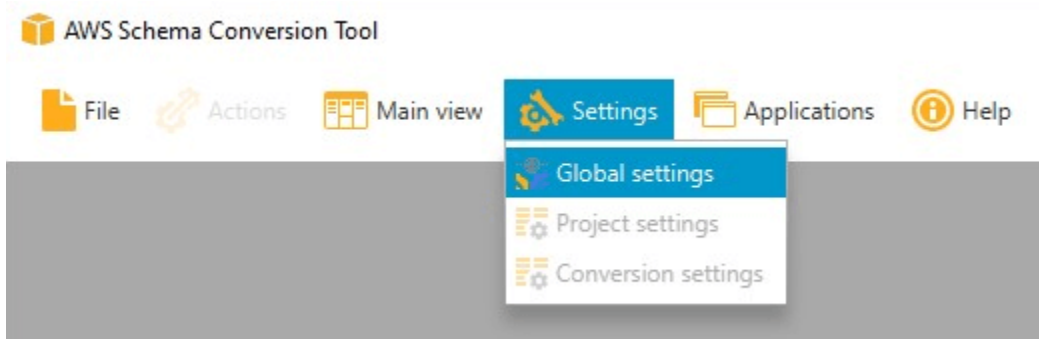
Moteur de base de données	Commandes d'installation
Microsoft SQL Server	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
MySQL	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Oracle	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo mkdir oracle-jdbc PROMPT&gt; cd oracle-jdbc PROMPT&gt; sudo cp -a /tmp/ojdbc8.jar .</pre>
PostgreSQL	<pre>PROMPT&gt; cd /usr/local/jdbc-drivers PROMPT&gt; sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

## Stockage des chemins des pilotes dans les paramètres globaux

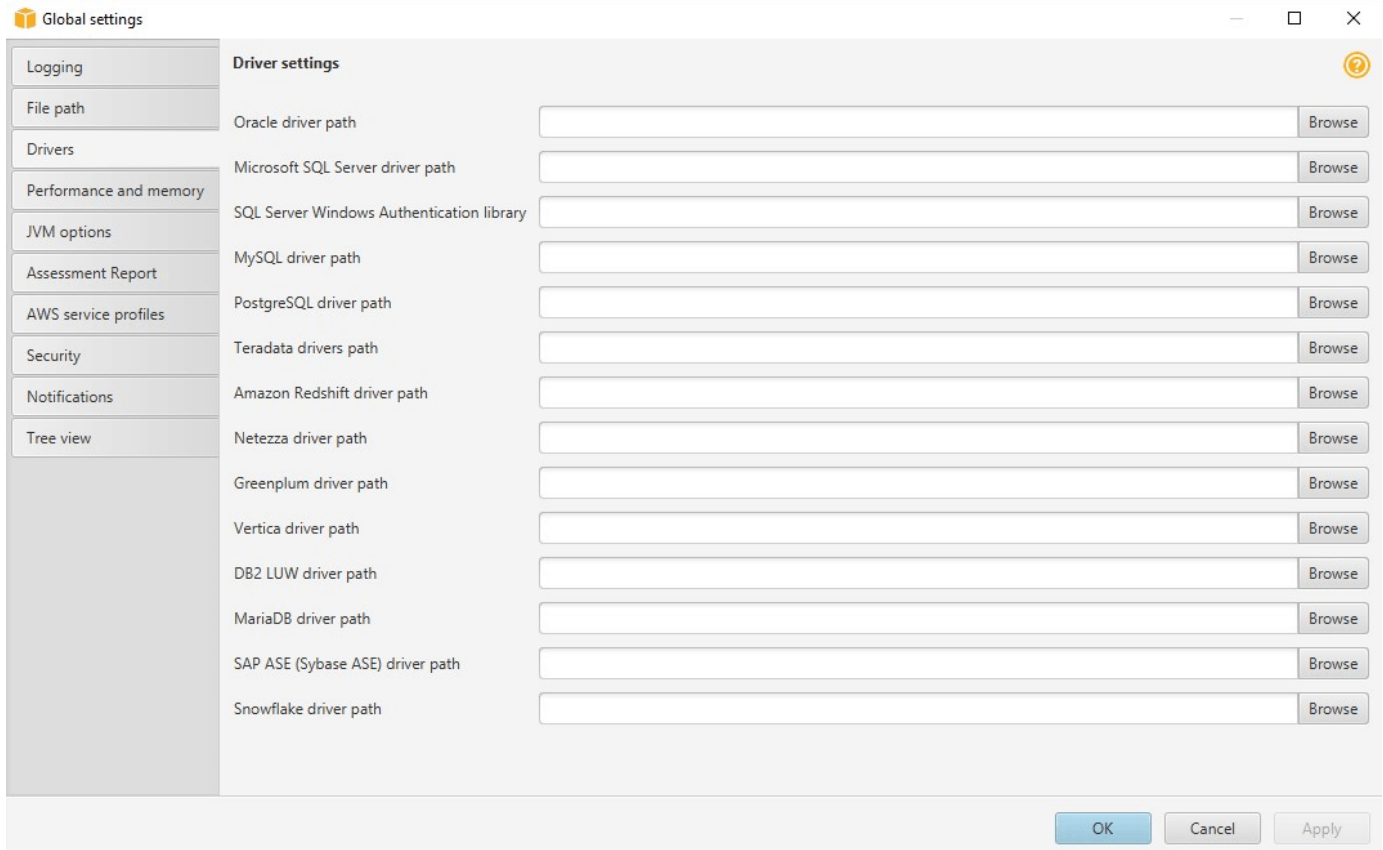
Après avoir téléchargé et installé les pilotes JDBC requis, vous pouvez définir l'emplacement global des pilotes dans les AWS SCT paramètres. Si vous ne définissez pas l'emplacement des pilotes de façon globale, l'application vous demandera l'emplacement des pilotes lorsque vous vous connecterez à une base de données.

Pour mettre à jour l'emplacement des fichiers de pilote

1. Dans AWS SCT, choisissez Paramètres, puis Paramètres généraux.



2. Dans Global settings (Paramètres globaux), choisissez Drivers (Pilotes). Ajoutez le chemin du fichier au pilote JDBC pour votre moteur de base de données source et le moteur de base de données de votre instance de base de données Amazon RDS cible.



3. Lorsque vous avez ajouté les chemins d'accès, cliquez sur OK.

## Mise à jour AWS Schema Conversion Tool

AWS mises à jour périodiques AWS SCT avec de nouvelles fonctionnalités. Si vous effectuez une mise à jour à partir d'une version précédente, créez un nouveau AWS SCT projet et reconvertissez tous les objets de base de données que vous utilisez.

Vous pouvez vérifier si des mises à jour existent pour AWS SCT.

Pour vérifier les mises à jour de AWS SCT

1. Lorsque vous êtes connecté AWS SCT, choisissez Aide, puis sélectionnez Vérifier les mises à jour.



2. Dans la boîte de dialogue Check for Updates (Rechercher les mises à jour), choisissez What's New (Nouveautés). Si le lien n'apparaît pas, cela signifie que vous disposez de la version la plus récente.

## AWS Schema Conversion Tool CLI

Vous pouvez télécharger la AWS SCT CLI pour une utilisation en ligne de commande. Pour télécharger le fichier JAR, cliquez sur le lien suivant :

[AWSSchemaConversionToolBatch.jar](#)

# Naviguer dans l'interface utilisateur du AWS SCT

Utilisez les rubriques suivantes pour vous aider à utiliser l'interface AWS SCT utilisateur. Pour plus d'informations sur l'installation AWS SCT, consultez [Installation et configuration AWS Schema Conversion Tool](#).

## Rubriques

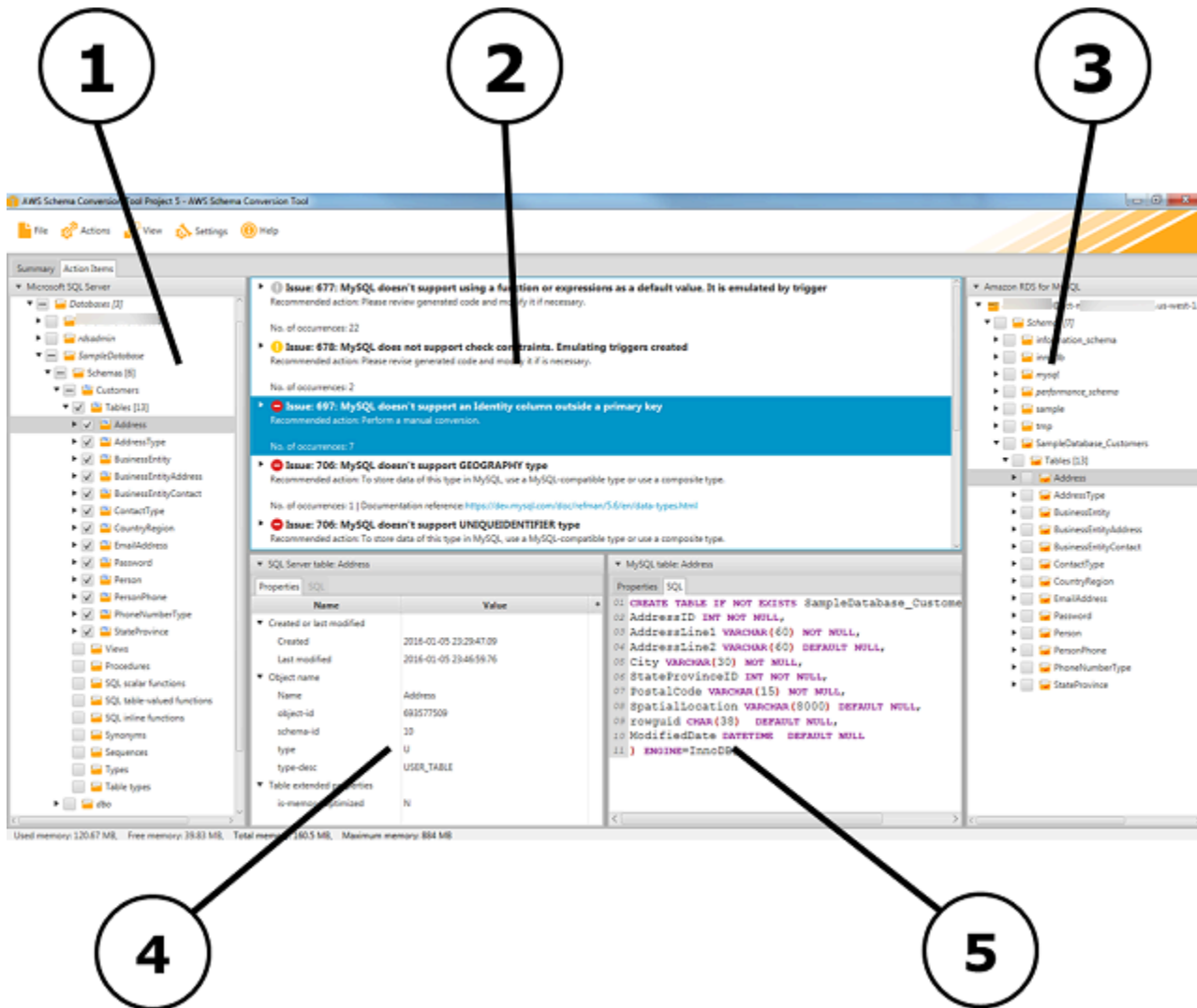
- [Affichage de la fenêtre du projet dans AWS SCT](#)
- [Démarrage et gestion de projets dans AWS SCT](#)
- [Utilisation de l' AWS SCT assistant](#)
- [Sauvegarde de projets dans AWS SCT](#)
- [Ajouter des serveurs au projet dans AWS SCT](#)
- [Utilisation du mode hors ligne dans AWS Schema Conversion Tool](#)
- [Utilisation de filtres arborescents dans AWS Schema Conversion Tool](#)
- [Masquer des schémas dans AWS Schema Conversion Tool](#)
- [Afficher le rapport d'évaluation dans AWS Schema Conversion Tool](#)
- [Conversion de schémas en AWS Schema Conversion Tool](#)
- [Appliquer les schémas convertis dans AWS Schema Conversion Tool](#)
- [Gestion des profils dans le AWS Schema Conversion Tool](#)
- [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#)
- [Stockage des mots de passe dans le AWS Schema Conversion Tool](#)
- [Création de la vue UNION ALL dans le AWS Schema Conversion Tool](#)
- [Utilisation des raccourcis clavier dans le AWS Schema Conversion Tool](#)

## Affichage de la fenêtre du projet dans AWS SCT

L'illustration suivante illustre ce que vous voyez AWS SCT lorsque vous créez un projet de migration de schéma, puis que vous convertissez un schéma.

1. Dans le volet gauche, le schéma de votre base de données source est présenté dans une arborescence. Votre schéma de base de données est « avec chargement différé ». En d'autres termes, lorsque vous sélectionnez un élément dans l'arborescence, vous obtenez et AWS SCT affiche le schéma actuel à partir de votre base de données source.

2. Dans le volet du milieu de la partie supérieure, les éléments de l'action apparaissent pour les éléments de schéma du moteur de base de données source qui n'a pas pu être converti automatiquement en moteur de base de données cible.
3. Dans le panneau droit, le schéma de votre instance DB cible est présenté dans une arborescence. Votre schéma de base de données est « avec chargement différé ». C'est-à-dire qu'au moment où vous sélectionnez un élément dans l'arborescence, vous obtenez et AWS SCT affiche le schéma actuel de votre base de données cible.



4. Dans le panneau inférieur gauche, lorsque vous choisissez un élément de schéma, les propriétés s'affichent. Ils décrivent l'élément du schéma source et la commande SQL permettant de créer cet élément dans la base de données source.

5. Dans le panneau inférieur droit, lorsque vous choisissez un élément de schéma, les propriétés s'affichent. Ils décrivent l'élément de schéma cible et la commande SQL permettant de créer cet élément dans la base de données cible. Vous pouvez modifier cette commande SQL et enregistrer la commande mise à jour avec votre projet.

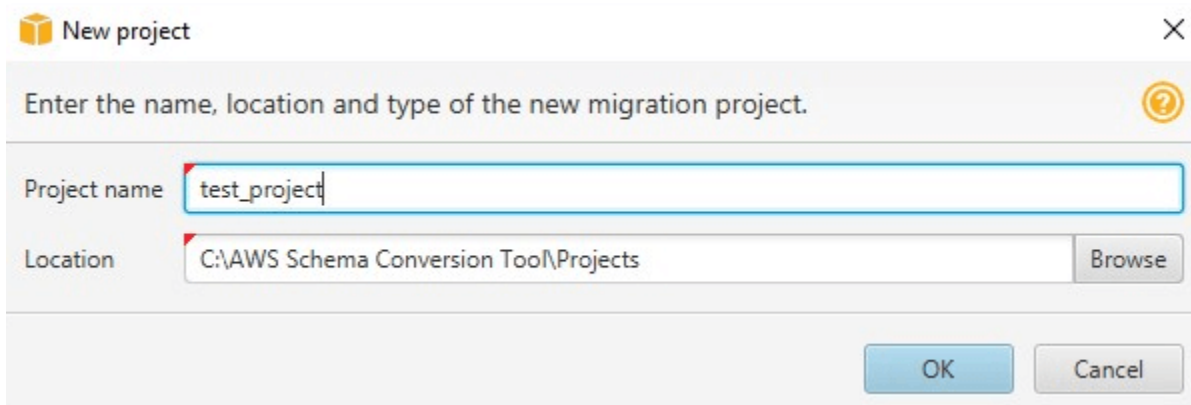
## Démarrage et gestion de projets dans AWS SCT

Pour démarrer AWS Schema Conversion Tool, double-cliquez sur l'icône de l'application.

Pour créer un AWS Schema Conversion Tool projet, procédez comme suit.

Pour créer votre projet

1. Démarrez le AWS Schema Conversion Tool.
2. Dans le menu Fichier, choisissez Nouveau projet. La boîte de dialogue Nouveau projet apparaît.



3. Entrez un nom pour votre projet, qui est stocké localement sur votre ordinateur.
4. Entrez l'emplacement de votre fichier de projet local.
5. Cliquez sur OK pour créer votre AWS SCT projet.
6. Choisissez Ajouter une source pour ajouter une nouvelle base de données source à votre AWS SCT projet. Vous pouvez ajouter plusieurs bases de données sources à votre AWS SCT projet.
7. Choisissez Ajouter une cible pour ajouter une nouvelle plateforme cible dans votre AWS SCT projet. Vous pouvez ajouter plusieurs plateformes cibles à votre AWS SCT projet.
8. Choisissez le schéma de base de données source dans le panneau de gauche.
9. Dans le panneau de droite, spécifiez la plate-forme de base de données cible pour le schéma source sélectionné.

10. Choisissez **Créer un mappage**. Ce bouton devient actif une fois que vous avez choisi le schéma de base de données source et la plate-forme de base de données cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données](#).

Votre AWS SCT projet est maintenant configuré. Vous pouvez enregistrer votre projet, créer un rapport d'évaluation de la migration de base de données et convertir les schémas de votre base de données source.

## Utilisation de l' AWS SCT assistant

Vous pouvez créer un nouveau projet de migration de base de données à l'aide du nouvel assistant de projet. Cet assistant vous aide à déterminer votre cible de migration et à vous connecter à vos bases de données. Il estime la complexité d'une migration pour toutes les destinations cibles prises en charge. Après avoir exécuté l'assistant, il AWS SCT produit un rapport récapitulatif pour la migration de votre base de données vers différentes destinations cibles. Vous pouvez utiliser ce rapport pour comparer les destinations cibles possibles et choisir le chemin de migration optimal.

Pour exécuter le nouvel assistant de projet

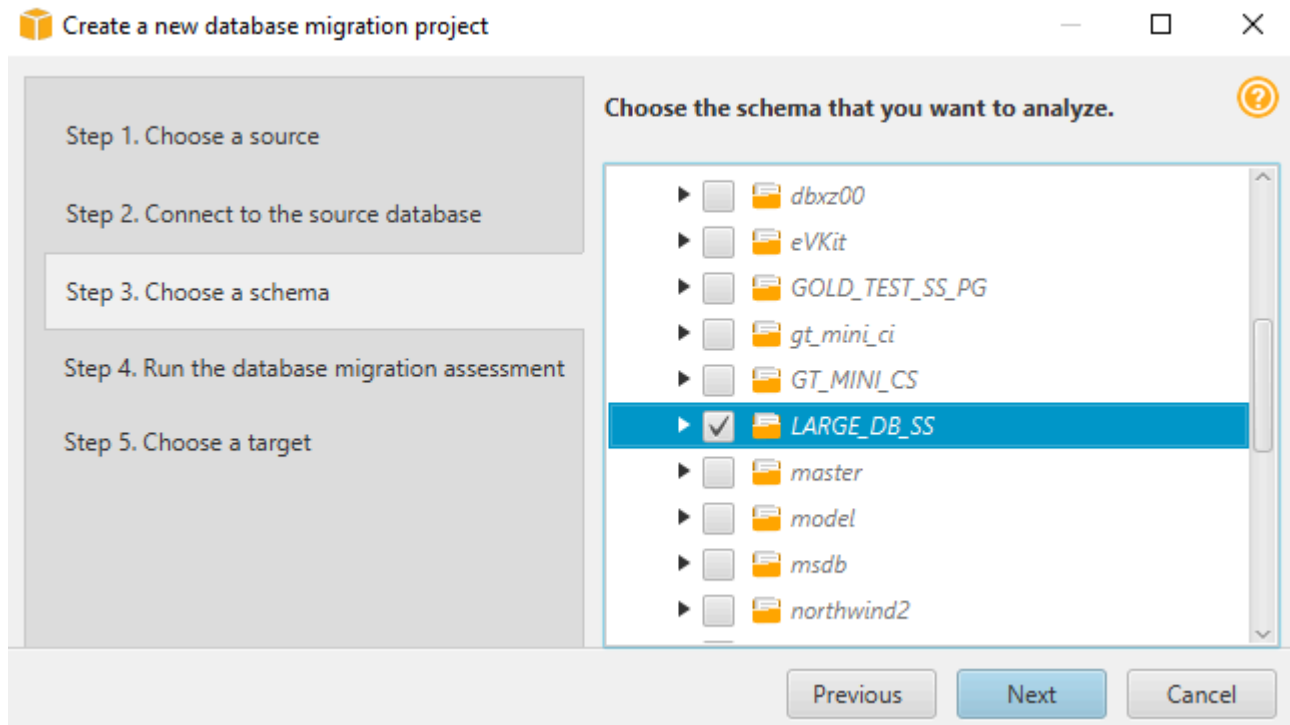
1. Choisissez votre base de données source.
  - a. Démarrez le AWS Schema Conversion Tool.
  - b. Dans le menu Fichier, choisissez **Nouvel assistant de projet**. La boîte de dialogue **Créer un nouveau projet de migration de base de données** s'ouvre.
  - c. Pour saisir les informations de connexion à la base de données source, suivez les instructions suivantes :

Paramètre	Action
Nom du projet	Entrez un nom pour votre projet, qui est stocké localement sur votre ordinateur.
Emplacement	Entrez l'emplacement de votre fichier de projet local.
Source type (Type de source)	Choisissez l'une des options suivantes : base de données SQL, base de données NoSQL ou ETL.

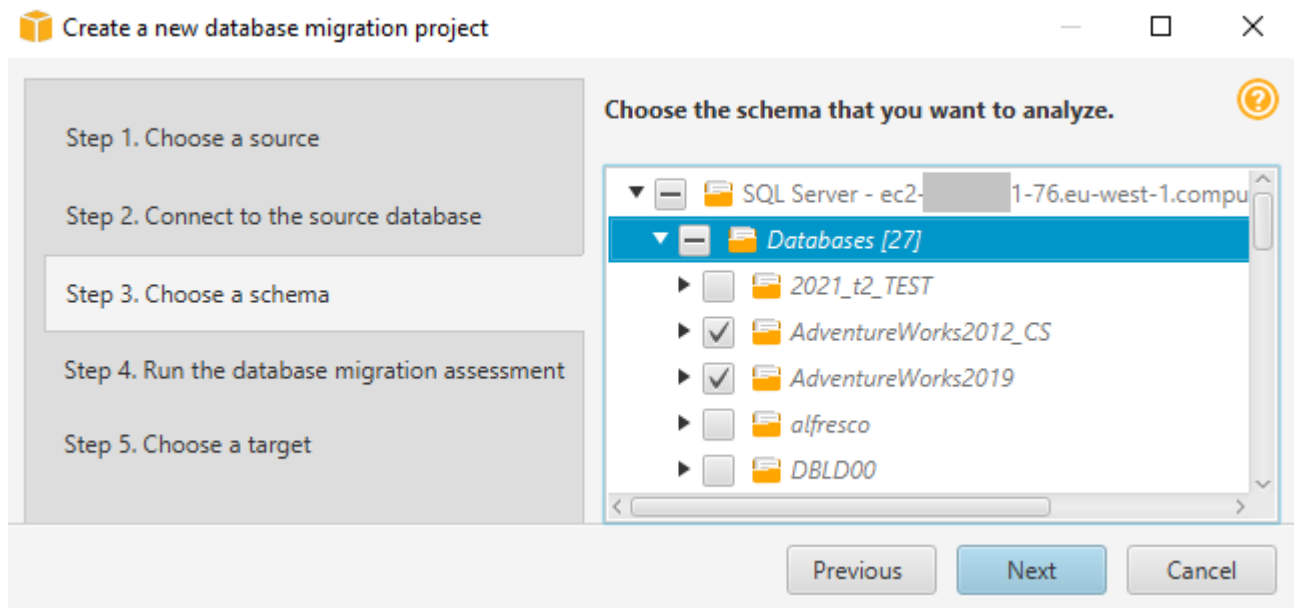
Paramètre	Action
	Si vous souhaitez consulter le rapport récapitulatif qui inclut toutes les destinations de migration, choisissez la base de données SQL.
Moteur source	Choisissez votre moteur de base de données source.
Stratégie de migration	<p>Choisissez l'une des options suivantes :</p> <ul style="list-style-type: none"><li>• Je souhaite changer de moteur et optimiser pour le cloud. Cette option convertit votre base de données source en un nouveau moteur de base de données.</li><li>• Je souhaite conserver le même moteur mais l'optimiser pour le cloud. Cette option permet de conserver votre moteur de base de données tel quel et de déplacer la base de données de l'environnement sur site vers le cloud.</li><li>• Je souhaite consulter un rapport combiné sur le changement de moteur de base de données et l'optimisation pour le cloud. Cette option compare la complexité de migration de toutes les options de migration disponibles.</li></ul> <p>Si vous souhaitez consulter le rapport d'évaluation agrégé qui inclut toutes les destinations de migration, choisissez la dernière option.</p>

- d. Choisissez Suivant. La page Connect to the source database (Connexion à la base de données source) s'ouvre.
2. Connectez-vous à votre base de données source.
    - a. Fournissez vos informations de connexion à la base de données source. Les paramètres de connexion dépendent de votre moteur de base de données source. Assurez-vous que l'utilisateur que vous utilisez pour l'analyse de votre base de données source dispose des autorisations applicables. Pour de plus amples informations, veuillez consulter [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#).
    - b. Choisissez Suivant. La page Choisir un schéma s'ouvre.
3. Choisissez le schéma de votre base de données.

- a. Cochez la case correspondant au nom des schémas que vous souhaitez évaluer, puis choisissez le schéma lui-même. Le nom du schéma est surligné en bleu lorsqu'il est sélectionné et le bouton Suivant est disponible.



- b. Si vous souhaitez évaluer plusieurs schémas de base de données, cochez les cases correspondant à tous les schémas, puis choisissez le nœud parent. Pour que l'évaluation soit réussie, vous devez choisir le nœud parent. Par exemple, pour une base de données SQL Server source, choisissez le nœud Bases de données. Le nom du nœud parent est surligné en bleu et le bouton Suivant est disponible.



- c. Choisissez Next. AWS SCT analyse les schémas de votre base de données source et crée un rapport d'évaluation de la migration de base de données. Le nombre d'objets de base de données dans les schémas de votre base de données source influe sur le temps nécessaire à l'exécution de l'évaluation. Lorsque vous avez terminé, la page Exécuter l'évaluation de la migration de base de données s'ouvre.
  4. Exécutez l'évaluation de la migration de la base de données.
    - a. Vous pouvez consulter et comparer les rapports d'évaluation pour différentes cibles de migration ou enregistrer une copie locale des fichiers de rapports d'évaluation pour une analyse plus approfondie.
    - b. Enregistrez une copie locale du rapport d'évaluation de la migration de la base de données. Choisissez Enregistrer, puis entrez le chemin d'accès au dossier dans lequel enregistrer les fichiers, puis cliquez sur Enregistrer. AWS SCT enregistre les fichiers du rapport d'évaluation dans le dossier spécifié.
    - c. Choisissez Suivant. La page Choisir une cible s'ouvre.
  5. Choisissez votre base de données cible.
    - a. Pour le moteur cible, choisissez le moteur de base de données cible que vous décidez d'utiliser en fonction du rapport d'évaluation.
    - b. Fournissez les informations de connexion à votre base de données cible. Les paramètres de connexion que vous voyez dépendent du moteur de base de données cible sélectionné. Assurez-vous que l'utilisateur indiqué pour la base de données cible dispose des autorisations requises. Pour plus d'informations sur les autorisations requises, consultez les sections qui



décrivent les autorisations pour les bases de données cibles dans [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#) et [Autorisations pour Amazon Redshift en tant que cible](#).

- c. Choisissez Terminer. AWS SCT crée votre projet et ajoute les règles de mappage. Pour de plus amples informations, veuillez consulter [Cartographie des types de données](#).

Vous pouvez maintenant utiliser le AWS SCT projet pour convertir les objets de votre base de données source.

## Sauvegarde de projets dans AWS SCT

Pour enregistrer un AWS Schema Conversion Tool projet, procédez comme suit.

Pour enregistrer votre projet

1. Démarrez le AWS Schema Conversion Tool.
2. Dans le menu Fichier, choisissez Enregistrer le projet.

AWS SCT enregistre le projet dans le dossier que vous avez spécifié lors de la création du projet.

Pour ouvrir un AWS Schema Conversion Tool projet existant, procédez comme suit.

Pour ouvrir votre projet

1. Dans le menu Fichier, choisissez Ouvrir un projet. La boîte de dialogue Ouvrir apparaît.
2. Choisissez le dossier du projet, puis le fichier Windows Script Component (\*.sct).
3. AWS SCT ouvre votre projet mais ne se connecte pas automatiquement à vos bases de données source et cible. Choisissez Connect to the server en haut de l'arborescence de votre schéma de base de données pour vous connecter à vos bases de données source et cible.

Si vous ouvrez un projet enregistré dans la AWS SCT version 1.0.655 ou antérieure, crée AWS SCT automatiquement des règles de mappage pour tous les schémas de base de données source vers la plate-forme de base de données cible. Pour ajouter d'autres plateformes de base de données cibles, supprimez les règles de mappage existantes, puis créez de nouvelles règles de mappage. Pour plus d'informations sur la création de règles de mappage, consultez [Cartographie des types de données](#).

# Ajouter des serveurs au projet dans AWS SCT

Vous pouvez ajouter plusieurs serveurs de base de données source et cible à un AWS Schema Conversion Tool projet.

Pour ajouter un serveur à votre projet

1. Démarrez le AWS Schema Conversion Tool.
2. Créez un nouveau projet ou ouvrez un projet existant.
3. Choisissez Ajouter une source dans le menu pour ajouter une nouvelle base de données source.
4. Choisissez une plate-forme de base de données et spécifiez les informations de connexion à la base de données. Pour plus d'informations sur la connexion à une base de données source, consultez [Connexion aux bases de données sources](#).

Pour vous connecter à votre base de données, procédez comme suit.

Pour vous connecter à votre base de données

1. Ouvrez le menu contextuel (clic droit) d'un serveur de base de données, puis choisissez Établir une connexion.

Vous pouvez également sélectionner Connect to the server en haut de l'arborescence du schéma de votre base de données.

2. Entrez le mot de passe pour vous connecter à votre serveur de base de données source.
3. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
4. Choisissez Connect pour vous connecter à votre base de données source.

Pour supprimer un serveur de base de données de votre AWS SCT projet, procédez comme suit.

Pour supprimer un serveur de base de données

1. Choisissez le serveur de base de données à supprimer.
2. Ouvrez le menu contextuel (clic droit), puis choisissez Supprimer du projet.

AWS SCT supprime le serveur de base de données sélectionné, toutes les règles de mappage, les résultats de conversion et les autres métadonnées associées à ce serveur.

# Utilisation du mode hors ligne dans AWS Schema Conversion Tool

Vous pouvez exécuter AWS Schema Conversion Tool en mode hors ligne. Vous découvrirez ci-après comment travailler avec un AWS SCT projet existant lorsque vous êtes déconnecté de votre base de données source.

AWS SCT ne nécessite pas de connexion à votre base de données source pour exécuter les opérations suivantes :

- Ajoutez des règles de mappage.
- Créez des rapports d'évaluation de la migration des bases de données.
- Convertissez les schémas et le code de base de données.
- Modifiez le code source et le code converti.
- Enregistrez votre code source et votre code converti sous forme de scripts SQL dans un fichier texte.

Avant de l'utiliser AWS SCT en mode hors ligne, connectez-vous à votre base de données source, chargez les métadonnées et enregistrez votre projet. Ouvrez ce projet ou déconnectez-vous du serveur de base de données source pour l'utiliser AWS SCT en mode hors ligne.

Pour exécuter AWS SCT en mode hors ligne

1. Démarrez le AWS Schema Conversion Tool et créez-en un nouveau. Pour de plus amples informations, veuillez consulter [Démarrage et gestion de projets dans AWS SCT](#).
2. Ajoutez un serveur de base de données source et connectez-vous à votre base de données source. Pour de plus amples informations, veuillez consulter [Ajouter des serveurs au projet dans AWS SCT](#).
3. Ajoutez un serveur de base de données cible ou utilisez une plate-forme de base de données cible virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).
4. Créez une règle de mappage pour définir la plate-forme de base de données cible pour votre base de données source. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).
5. Choisissez Affichage, puis sélectionnez Affichage principal.
6. Dans le panneau de gauche qui affiche les objets de votre base de données source, choisissez les schémas de votre base de données source. Ouvrez le menu contextuel (clic droit) de l'objet,

puis choisissez **Charger le schéma**. Cette opération charge toutes les métadonnées du schéma source dans votre AWS SCT projet.

Les opérations de création de rapport et de conversion de schéma chargent également toutes les métadonnées du schéma source dans votre AWS SCT projet. Si vous avez exécuté l'une de ces opérations depuis le menu contextuel, ignorez l'opération **Charger le schéma**.

7. Dans le menu **Fichier**, choisissez **Enregistrer le projet** pour enregistrer les métadonnées de la base de données source dans votre projet.
8. Choisissez **Déconnecter du serveur** pour vous déconnecter de votre base de données source. Vous pouvez maintenant l'utiliser AWS SCT en mode hors ligne.

## Utilisation de filtres arborescents dans AWS Schema Conversion Tool

Pour migrer des données d'une source vers une cible, AWS SCT charge toutes les métadonnées des bases de données source et cible dans une arborescence. Cette structure s'affiche dans AWS SCT sous la forme d'une vue d'arborescence dans la fenêtre de projet principale.

Certaines bases de données peuvent comporter un grand nombre d'objets dans la structure d'arborescence. Vous pouvez utiliser des filtres d'arborescence AWS SCT pour rechercher des objets dans les structures arborescentes source et cible. Lorsque vous utilisez un filtre d'arborescence, vous ne changez pas les objets qui sont convertis lorsque vous convertissez votre base de données. Le filtre modifie uniquement ce que vous voyez dans l'arborescence.

Les filtres arborescents fonctionnent avec les AWS SCT objets préchargés. En d'autres termes, AWS SCT ne permet pas de charger des objets à partir de la base de données durant les recherches. Cette approche signifie que l'arborescence contient généralement moins d'objets que la base de données n'en contient.

Tenez compte des éléments suivants concernant les filtres d'arborescence :

- Le filtre par défaut est **ANY**, ce qui signifie que le filtre utilise un nom de recherche pour trouver des objets.
- Lorsque vous sélectionnez un ou plusieurs types d'objets, vous voyez uniquement ces types d'objets dans l'arborescence.

- Vous pouvez utiliser le masque de filtre pour afficher différents types de symboles, y compris les caractères Unicode, les espaces et les caractères spéciaux. Le caractère « % » est le caractère générique pour tout symbole.
- Une fois que vous avez appliqué un filtre, le décompte indique uniquement le nombre d'objets filtrés.

### Pour créer un filtre d'arborescence

1. Ouvrez un AWS SCT projet existant.
2. Connectez-vous à la base de données à laquelle vous souhaitez appliquer le filtre d'arborescence.
3. Choisissez cette icône.




L'icône d'annulation de filtre est grisée, car aucun filtre n'est actuellement appliqué.




4. Entrez les informations suivantes dans la boîte de dialogue Filtre. Les options de la boîte de dialogue sont différentes pour chaque moteur de base de données.

AWS SCT option de filtre	Action
Niveau	<p>Choisissez Catégories pour filtrer les objets par catégories.</p> <p>Choisissez Statuts pour filtrer les objets par statut.</p>
Type	<p>Pour Catégories dans Level, choisissez les catégories d'objets filtrés. Choisissez Any loaded pour afficher les objets de toutes les catégories.</p> <p>Pour les statuts dans Level, choisissez le statut des objets filtrés. Vous pouvez choisir l'une des options suivantes :</p> <ul style="list-style-type: none"> <li>• Converti pour afficher tous les objets convertis</li> <li>• Dispose d'actions pour afficher tous les objets présentant des problèmes de conversion</li> </ul>

AWS SCT option de filtre	Action
	<ul style="list-style-type: none"> <li>• Chiffré pour afficher tous les objets chiffrés</li> </ul>
Condition	<p>Pour les catégories du niveau, choisissez la condition de filtrage entre J'aime et Je n'aime pas.</p> <p>Pour les statuts dans Level, l'option de condition de filtrage n'est pas disponible.</p>
Valeur	<p>Pour Catégories dans Level, entrez la valeur pour filtrer l'arborescence en fonction de cette valeur.</p> <p>Utilisez le pourcentage (%) comme caractère générique pour afficher tous les objets.</p> <p>Pour les statuts dans Level, choisissez la valeur entre Vrai et Faux.</p>
Et/ou	<p>Choisissez AND ou des opérateurs OR logiques pour appliquer plusieurs clauses de filtre.</p>

 Filter
×

Specify multiple filters or filter values for schemas or any other objects. Use % as a wildcard. ?

			Level	Type	Condition	Value	And/Or
+		<input type="checkbox"/>	Categories ▾	Any loaded ▾	Like ▾	<input type="text" value="%dbo%"/>	AND ▾
+		<input type="checkbox"/>	Categories ▾	Tables ▾	Like ▾	<input type="text" value="%tmp%"/>	AND ▾
+		<input type="checkbox"/>	Statuses ▾	Mapped ▾	Value	<input type="text" value="True"/>	▾

Any loaded like %dbo% AND Tables like %tmp% AND mapped value true

5. Choisissez Ajouter une nouvelle clause pour ajouter une clause de filtre supplémentaire. AWS SCT peut appliquer plusieurs clauses de filtre à l'aide AND d'opérateurs OR logiques.
6. Choisissez Appliquer. Une fois que vous avez choisi Apply, l'icône d'annulation de filtre (en regard de l'icône de filtre) est activée. Utilisez cette icône si vous souhaitez supprimer les filtres que vous avez appliqués.
7. Choisissez Close (Fermer) pour fermer la boîte de dialogue.

Lorsque vous filtrez le schéma qui apparaît dans l'arborescence, vous ne changez pas les objets qui sont convertis lorsque vous convertissez votre schéma. Le filtre change uniquement ce que vous voyez dans l'arborescence.

## Importation d'une liste de fichiers pour le filtre d'arborescence

Vous pouvez importer un fichier de valeurs séparées par des virgules (CSV) avec des points-virgules ou un fichier JSON contenant les noms ou les valeurs que vous souhaitez que le filtre d'arborescence utilise. Ouvrez un AWS SCT projet existant, connectez-vous à la base de données à laquelle appliquer le filtre d'arborescence, puis choisissez l'icône du filtre.

Pour télécharger un exemple du fichier, choisissez Télécharger le modèle. Entrez le nom du fichier et choisissez Enregistrer.

Pour télécharger vos paramètres de filtre existants, choisissez Exporter. Entrez le nom du fichier et choisissez Enregistrer.

Pour importer une liste de fichiers pour le filtre d'arborescence, choisissez Importer. Choisissez un fichier à importer, puis choisissez Open. Choisissez Apply, puis Close.

Les fichiers CSV utilisent un point-virgule comme séparateur et ont le format suivant :

- `object_type` est le type d'objet que vous voulez rechercher.
- `database_name` est le nom de la base de données dans laquelle se trouve cet objet.
- `schema_name` est le nom du schéma dans lequel se trouve cet objet.
- `object_name` est le nom de l'objet.
- `import_type` spécifie à `include` ou `exclude` cet élément à partir du filtre.

Utilisez des fichiers JSON pour décrire les cas de filtrage complexes, tels que les règles imbriquées. Les fichiers JSON ont le format suivant :

- `filterGroupType` est le type de règle de filtre (AND ou d'opérateurs OR logiques) qui s'applique à plusieurs clauses de filtre.
- `filterCategory` est le niveau du filtre (catégories ou statuts).
- `names` est la liste des noms d'objets qui s'appliquent au filtre Catégories.
- `filterCondition` est la condition de filtrage (LIKE ou NOT LIKE) qui s'applique au filtre Catégories.
- `transformName` est le nom de statut qui s'applique au filtre d'état.
- `value` est la valeur à utiliser pour filtrer l'arbre.
- `transformValue` est la valeur du filtre (TRUE ou FALSE) qui s'applique au filtre d'état.

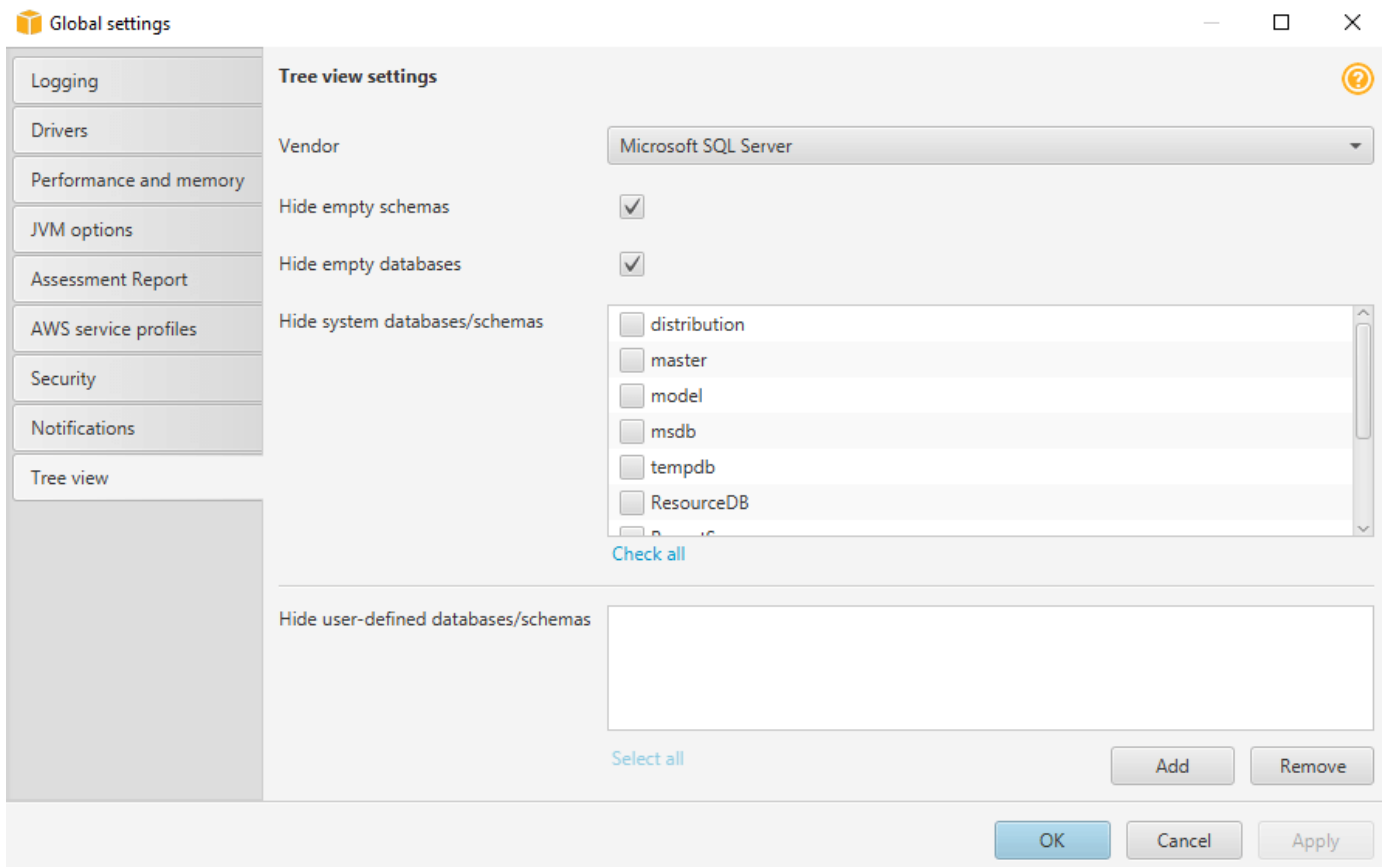
## Masquer des schémas dans AWS Schema Conversion Tool

Utilisez les paramètres de l'arborescence pour spécifier les schémas et les bases de données que vous souhaitez voir apparaître dans l' AWS SCT arborescence. Vous pouvez masquer les schémas vides, les bases de données vides, les bases de données système, et les bases de données et schémas définis par l'utilisateur.

Pour masquer les bases de données et les schémas dans la vue d'arborescence

1. Ouvrez un AWS SCT projet.
2. Connectez-vous à la banque de données que vous souhaitez afficher dans la vue d'arborescence.
3. Choisissez Paramètres, Paramètres généraux, Vue en arborescence.





4. Dans la section Paramètres de l'affichage en arborescence, procédez comme suit :
  - Pour le fournisseur, choisissez la plate-forme de base de données.
  - Choisissez Masquer les schémas vides pour masquer les schémas vides pour la plate-forme de base de données sélectionnée.
  - Choisissez Masquer les bases de données vides pour masquer les bases de données vides pour la plate-forme de base de données sélectionnée.
  - Pour Masquer les bases de données/schémas système, choisissez les bases de données système et les schémas par nom pour les masquer.
  - Pour Masquer les bases de données/schémas définis par l'utilisateur, entrez les noms des bases de données définies par l'utilisateur et des schémas que vous souhaitez masquer, puis choisissez Ajouter. Ces noms ne sont pas sensibles à la casse.
5. Choisissez OK.

# Afficher le rapport d'évaluation dans AWS Schema Conversion Tool

Le rapport d'évaluation de la migration de base de données résume toutes les mesures à prendre pour les schémas qui ne peuvent pas être convertis automatiquement dans le moteur de votre instance de base de données Amazon RDS cible. Le rapport donne également une estimation du temps nécessaire pour écrire le code équivalent pour l'instance DB cible.

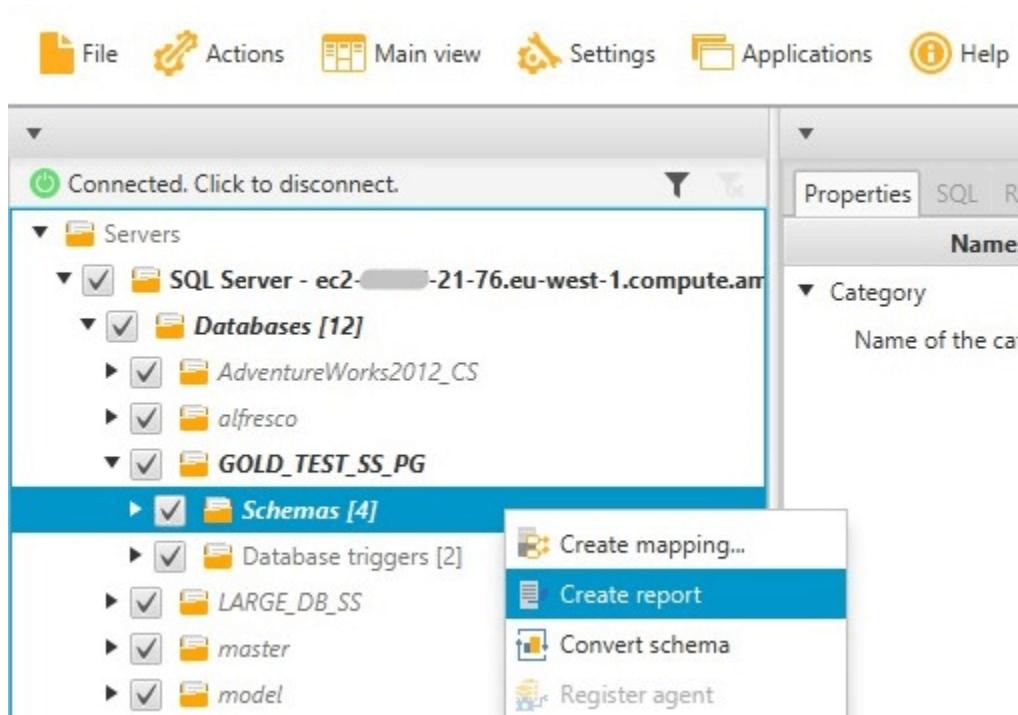
Vous pouvez créer un rapport d'évaluation de la migration de base de données après avoir ajouté les bases de données source et les plateformes cibles à votre projet et avoir défini des règles de mappage.

Pour créer et afficher le rapport d'évaluation de migration de base de données

1. Assurez-vous d'avoir créé une règle de mappage pour le schéma de base de données source afin de créer un rapport d'évaluation pour. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
2. Dans le menu Affichage, choisissez Vue principale.
3. Dans le panneau de gauche qui affiche le schéma de votre base de données source, choisissez les objets du schéma pour lesquels créer un rapport d'évaluation.

Assurez-vous d'avoir coché les cases correspondant à tous les objets de schéma pour lesquels créer un rapport d'évaluation.

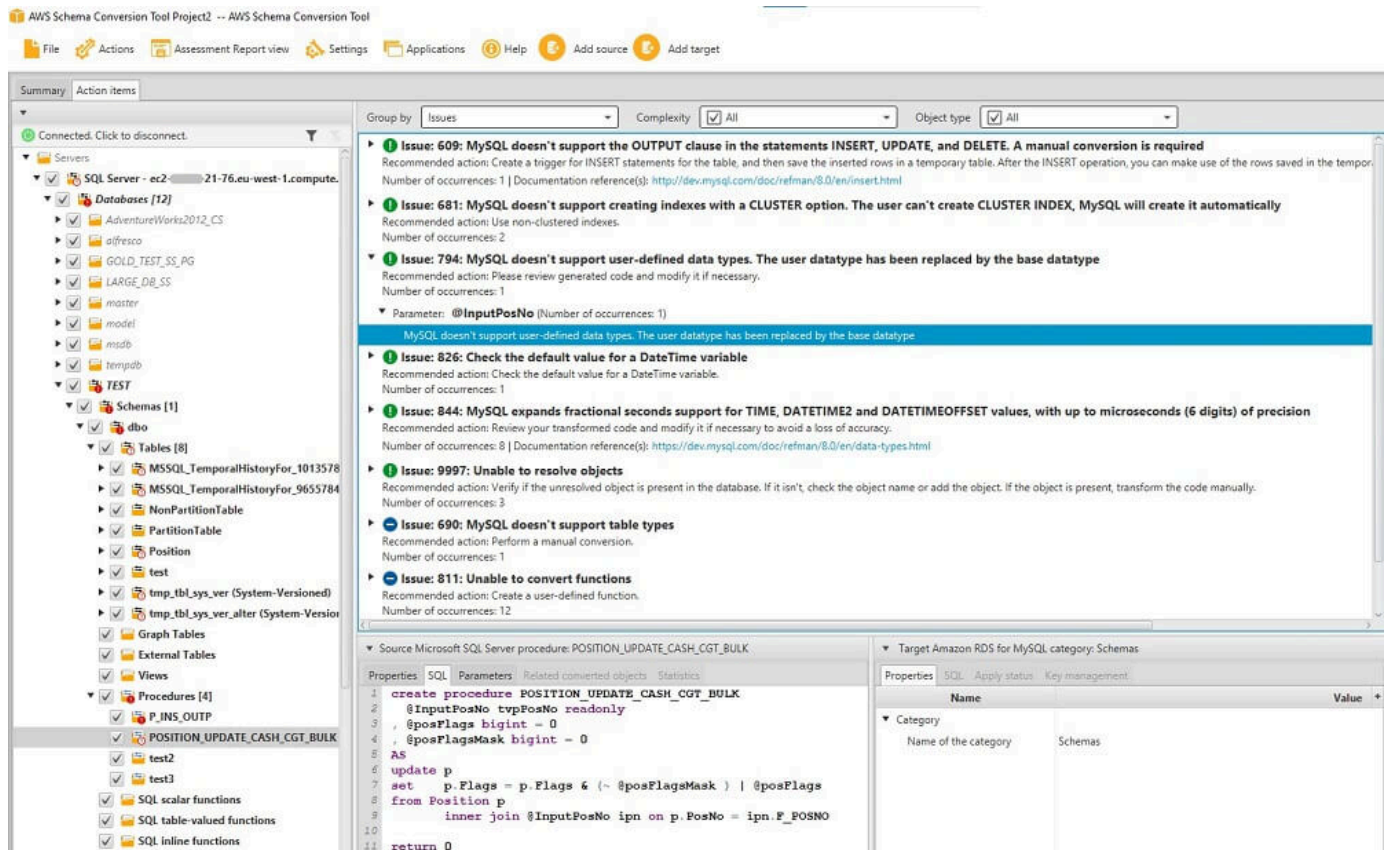
4. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Créer un rapport.



La vue du rapport d'évaluation s'ouvre.

5. Choisissez l'onglet Actions.

L'onglet Actions affiche une liste d'éléments décrivant le schéma qui ne peut pas être converti automatiquement. Choisissez l'une des actions de la liste. AWS SCT met en évidence l'élément de votre schéma auquel l'action s'applique, comme indiqué ci-dessous.



## 6. Choisissez l'onglet Summary.

L'onglet Summary affiche un récapitulatif du rapport d'évaluation de migration de base de données. Il indique le nombre d'éléments qui ont été convertis automatiquement et le nombre de ceux qui ne l'ont pas été. Ce récapitulatif comprend en outre une estimation du temps que prendra la création d'un schéma dans l'instance DB cible équivalent à celui de la base de données source.

La section Évaluation des licences et support cloud contient des informations sur le déplacement de votre schéma de base de données sur site existant vers une instance de base de données Amazon RDS exécutant le même moteur. Par exemple, si vous souhaitez modifier les types de licence, cette section du rapport indique les fonctionnalités à supprimer de votre base de données actuelle.

Un exemple de récapitulatif de rapport d'évaluation est illustré ci-après.

Summary | Action Items

Save to CSV Save to PDF

## Database migration assessment report

Source database: GOLD\_TEST\_SS\_PG-21-76.eu-west-1.compute.amazonaws.com/GOLD\_TEST\_SS\_PG:1433  
 Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (X64) Mar 22 2021 18:10:24  
 Copyright (C) 2019 Microsoft Corporation  
 Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763:) (Hypervisor)  
 Case sensitivity: Off

### Executive summary

We completed the analysis of your Microsoft SQL Server source database and estimate that 90% of the database storage objects and 77% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for PostgreSQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, table types, sequences, synonyms and xml schema collections. Database code objects include triggers, views, procedures, scalar functions, inline functions, table-valued functions and database triggers. Based on the source code syntax analysis, we estimate 94% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically. To complete the migration, we recommend 3,300 conversion action(s) ranging from simple tasks to medium-complexity actions to complex conversion actions.

Migration guidance for database objects that could not be converted automatically can be found [here](#)

### Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

**Figure: Conversion statistics for database storage objects**

Object Type	Count	Automatically converted	Simple actions	Medium-complexity actions	Complex actions
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	2%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	35%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	80%	0%	0%

7. Choisissez l'onglet Summary, puis Save to PDF. Le rapport d'évaluation de migration de base de données est enregistré au format PDF. Le fichier PDF contient à la fois le récapitulatif et les informations sur les éléments d'action.

Vous pouvez également choisir Enregistrer au format CSV pour enregistrer le rapport sous forme de fichier CSV. Lorsque vous choisissez cette option, AWS SCT trois fichiers CSV sont créés. Ces fichiers contiennent les informations suivantes :

- Liste des actions de conversion avec des actions recommandées.
- Récapitulatif des éléments d'action de conversion avec une estimation de l'effort requis pour convertir une occurrence de l'élément d'action.
- Un résumé présentant un certain nombre d'actions classées en fonction du temps estimé de conversion.

## Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

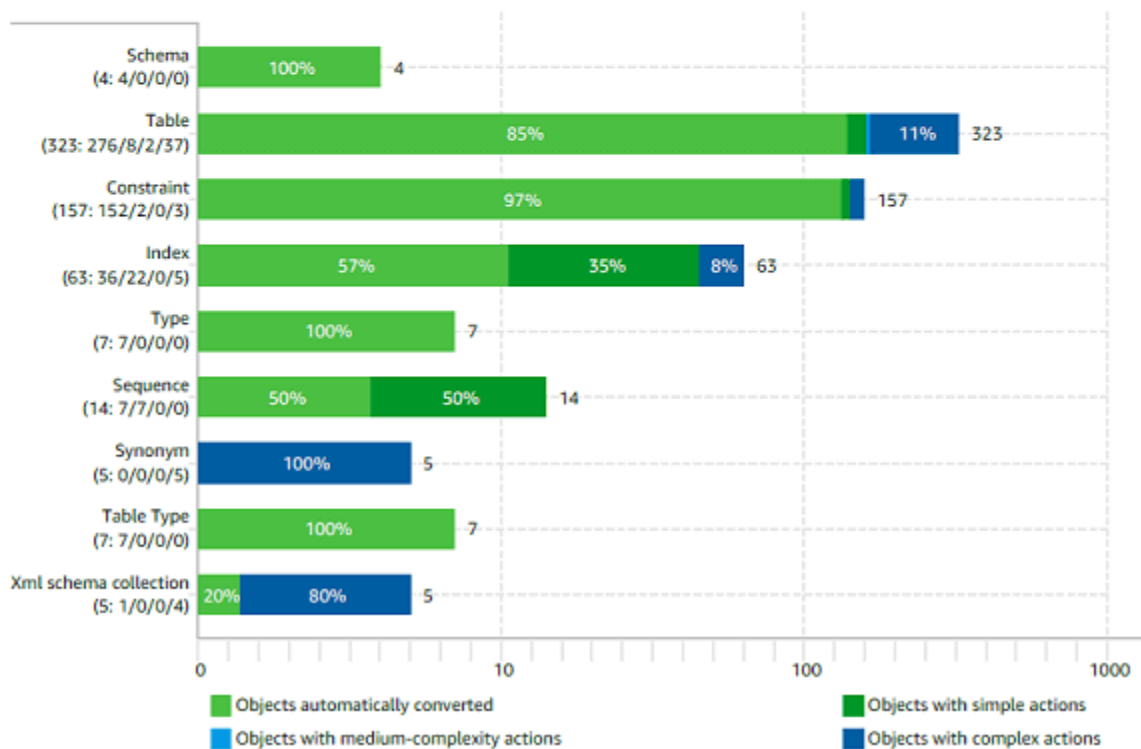
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects

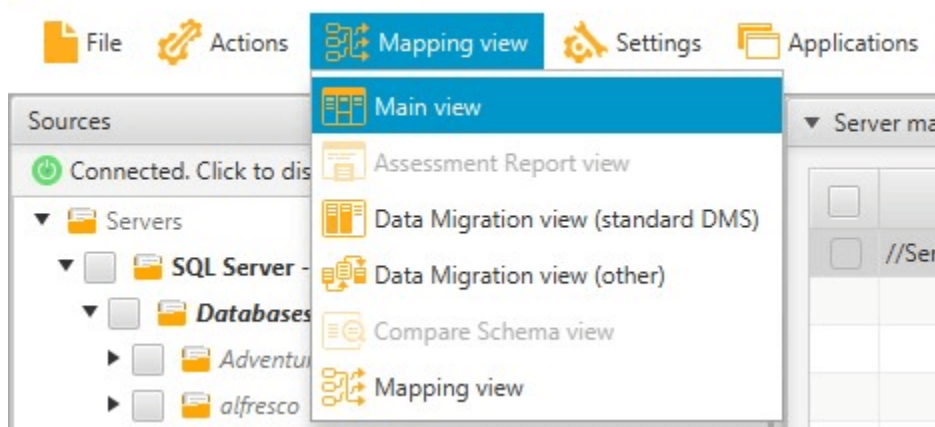


## Conversion de schémas en AWS Schema Conversion Tool

Après avoir ajouté des bases de données source et cible à votre projet et créé des règles de mappage, vous pouvez convertir les schémas de vos bases de données source. Utilisez la procédure suivante pour convertir un schéma.

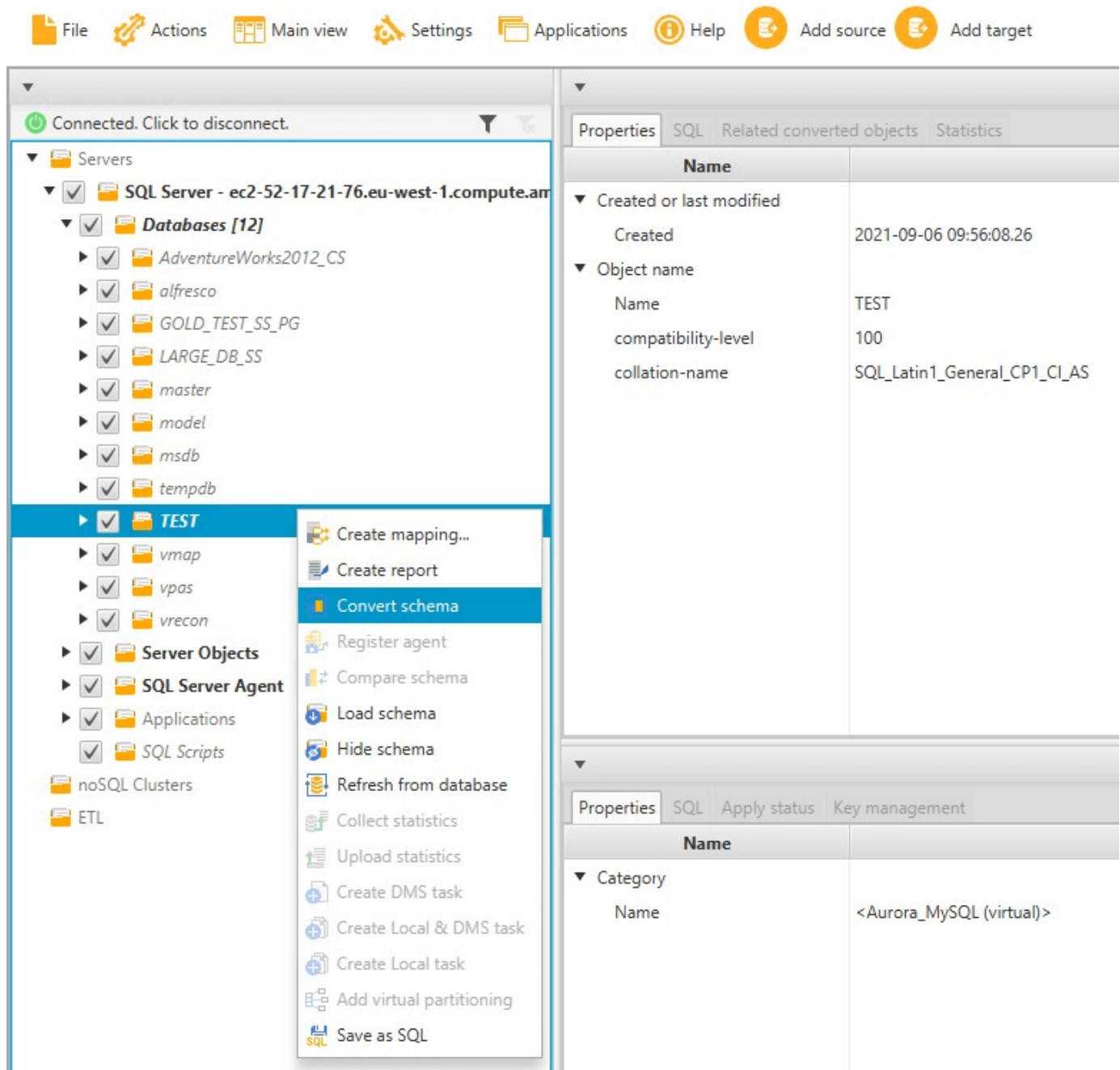
## Pour convertir votre schéma

1. Choisissez Affichage, puis sélectionnez Affichage principal.



2. Dans le panneau de gauche qui affiche le schéma de votre base de données source, cochez la case correspondant au nom de l'objet à convertir. Choisissez ensuite cet objet. AWS SCT met en évidence le nom de l'objet en bleu. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Convertir le schéma.

Pour convertir plusieurs objets de base de données, cochez les cases correspondant à tous les objets. Choisissez ensuite le nœud parent. Par exemple, pour les tables, le nœud parent est Tables. Assurez-vous que le nom du nœud parent est AWS SCT surligné en bleu. Ouvrez le menu contextuel (clic droit) du nœud parent, puis choisissez Convertir le schéma.

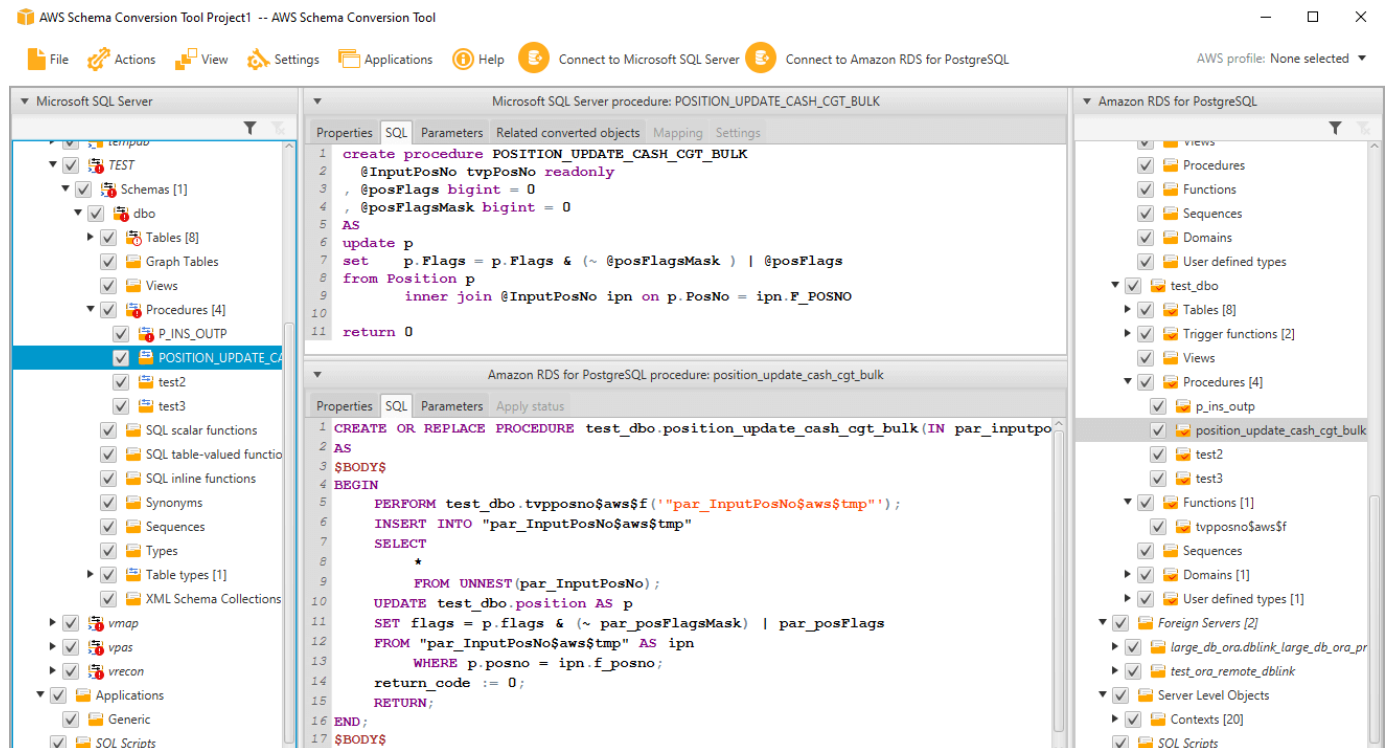


3. Lorsque la conversion du schéma est AWS SCT terminée, vous pouvez consulter le schéma proposé dans le panneau situé à droite de votre projet.

À ce stade, aucun schéma n'est appliqué à votre instance de base de données cible. Le schéma planifié est un composant du projet. Si vous choisissez un élément de schéma converti, vous pouvez voir la commande de schéma planifiée dans le panneau situé en bas au centre pour votre instance de base de données cible.



Vous pouvez modifier le schéma dans cette fenêtre. Le schéma modifié est stocké dans le cadre de votre projet et est écrit dans l'instance de base de données cible lorsque vous choisissez d'appliquer votre schéma converti.



## Appliquer les schémas convertis dans AWS Schema Conversion Tool

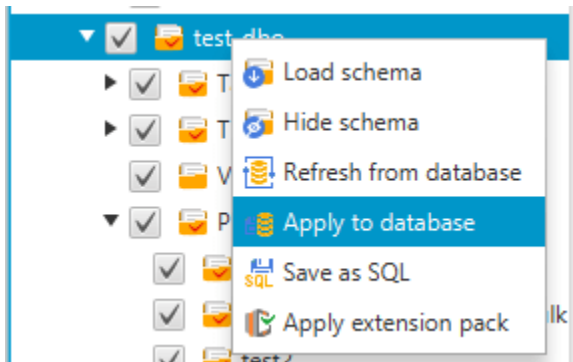
Vous pouvez appliquer le schéma de base de données converti à votre instance de base de données cible. Une fois que le schéma a été appliqué à l'instance DB cible, vous pouvez le mettre à jour en fonction des éléments d'action figurant dans le rapport d'évaluation de migration de base de données.

### Warning

La procédure suivante remplace le schéma cible existant. Veillez à ne pas remplacer les schémas par inadvertance. Veillez à ne pas remplacer les schémas de votre instance de base de données cible que vous avez déjà modifiés, ou à ne pas remplacer ces modifications.

Pour appliquer le schéma de base de données converti à votre instance de base de données cible

1. Choisissez **Connect to the server** en haut du panneau de droite de votre projet pour vous connecter à votre base de données cible. Si vous êtes connecté à votre base de données cible, ignorez cette étape.
2. Choisissez l'élément de schéma dans le panneau droit de votre projet, qui présente le schéma planifié de l'instance DB cible.
3. Ouvrez le menu contextuel (clic droit) pour l'élément de schéma, puis choisissez **Apply to database**.



Le schéma converti est appliqué à l'instance DB cible.

## Gestion des profils dans le AWS Schema Conversion Tool

Vous pouvez enregistrer vos AWS informations d'identification dans AWS SCT. AWS SCT utilise vos informations d'identification lorsque vous utilisez des fonctionnalités intégrées aux AWS services. Par exemple, AWS SCT s'intègre à Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS), AWS Database Migration Service et AWS DMS().

AWS SCT vous demande vos AWS informations d'identification lorsque vous accédez à une fonctionnalité qui les requiert. Vous pouvez stocker vos informations d'identification dans les paramètres globaux de l'application. Lorsque vous êtes invité à saisir vos informations d'identification, vous pouvez sélectionner les informations d'identification stockées.

Vous pouvez enregistrer différents ensembles d'AWS informations d'identification dans les paramètres globaux de l'application. Par exemple, il est possible de stocker un ensemble d'informations d'identification que vous utilisez dans les scénarios de test et un second, utilisé dans les scénarios de production. Vous pouvez également stocker différentes informations d'identification pour différentes Régions AWS.

## Stockage des AWS identifiants

Utilisez la procédure suivante pour stocker les informations d' AWS identification dans le monde entier.

Pour stocker les AWS informations d'identification

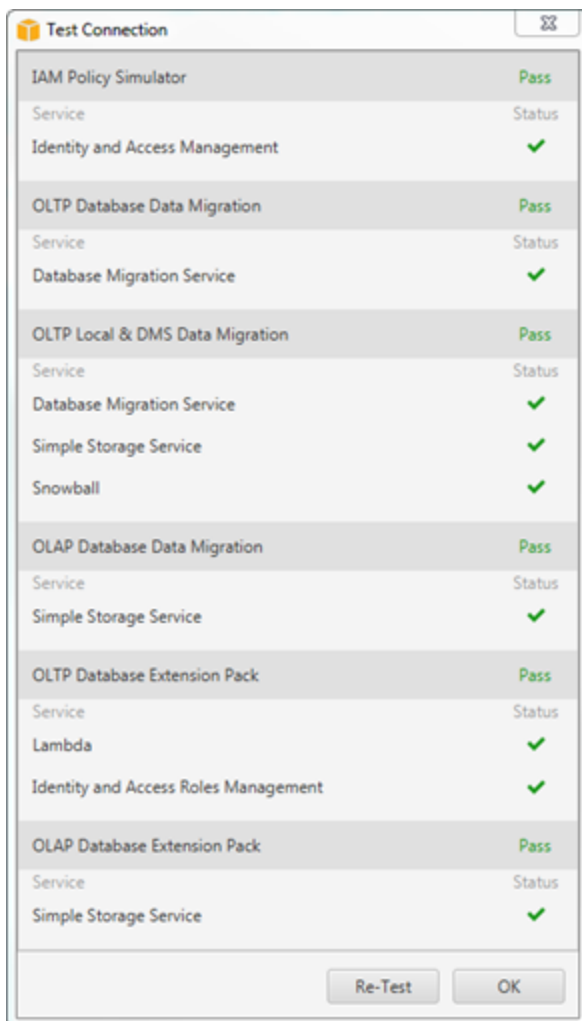
1. Démarrez le AWS Schema Conversion Tool.
2. Ouvrez le menu Paramètres, puis choisissez Paramètres globaux. La boîte de dialogue Global settings s'affiche.
3. Choisissez des profils de AWS service, puis choisissez Ajouter un nouveau profil AWS de service.
4. Entrez vos AWS informations comme suit.

AWS SCT option	Action
Profile name (Nom de profil)	Entrez un nom pour votre profil.
AWS clé d'accès	Entrez votre clé AWS d'accès.
AWS clé secrète	Entrez votre clé d'accès AWS secrète. Pour plus d'informations sur les clés AWS d'accès, consultez <a href="#">la section Gestion des clés d'accès</a> dans le guide de l'utilisateur IAM.
Région	Choisissez le Région AWS pour votre profil.
Dossier de compartiment Amazon S3	Choisissez le compartiment Amazon S3 correspondant à votre profil. Vous devez spécifier un compartiment uniquement si vous utilisez une fonctionnalité qui se connecte à Amazon S3. Pour plus d'informations sur les privilèges requis, consultez <a href="#">Autorisations d'utilisation du profil AWS de service</a> .

Choisissez Use FIPS endpoint for S3 si vous avez besoin de respecter les exigences de sécurité de la norme FIPS (Federal Information Processing Standard). Les points de terminaison FIPS sont disponibles dans les régions suivantes : AWS

- US East (N. Virginia) Region
  - US East (Ohio) Region
  - Région US West (N. California)
  - Région US West (Oregon)
5. Choisissez Tester la connexion pour vérifier que vos informations d'identification sont correctes et actives.

La boîte de dialogue Tester la connexion apparaît. Vous pouvez voir le statut de chacun des services connectés à votre profil. Pass indique que le profil peut accéder au service.



6. Une fois que vous avez configuré votre profil, choisissez Save pour enregistrer votre profil ou Cancel pour annuler vos modifications.
7. Cliquez sur OK pour fermer la boîte de dialogue des paramètres généraux.

## Définition du profil par défaut pour un projet

Vous pouvez définir le profil par défaut d'un AWS SCT projet. Cela permet d'associer les AWS informations d'identification stockées dans le profil au projet. Le projet étant ouvert, procédez comme suit pour définir le profil par défaut.

Pour définir le profil par défaut pour un projet

1. Démarrez le AWS Schema Conversion Tool et créez-en un nouveau.
2. Dans le menu Paramètres, choisissez Paramètres du projet. La boîte de dialogue des paramètres du projet apparaît.
3. Choisissez l'onglet Environnement du projet.
4. Choisissez Ajouter un nouveau profil de AWS service pour ajouter un nouveau profil. Ensuite, pour le profil de AWS service, choisissez le profil que vous souhaitez associer au projet.
5. Cliquez sur OK pour fermer la boîte de dialogue des paramètres du projet. Vous pouvez également choisir Cancel pour annuler vos modifications.

## Autorisations d'utilisation du profil AWS de service

Les autorisations suivantes sont requises pour accéder à votre compartiment Amazon S3 depuis votre profil de AWS service :

- `s3:PutObject`— pour ajouter des objets dans votre compartiment Amazon S3.
- `s3:DeleteObject`— pour supprimer la version nulle d'un objet et insérer un marqueur de suppression, qui devient la version actuelle de l'objet.
- `s3:ListBucket`— pour renvoyer jusqu'à 1 000 objets depuis votre compartiment Amazon S3.
- `s3:GetObject`— pour récupérer des objets depuis votre compartiment Amazon S3.

L'exemple de code suivant vous montre comment accorder ces autorisations à votre utilisateur.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
```

```
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "*"
    ]
}
]
```

## Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool

AWS SCT peut utiliser les informations d'identification de base de données que vous stockez dans AWS Secrets Manager. Vous pouvez renseigner toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager. Pour utiliser Secrets Manager, assurez-vous de stocker AWS les profils dans le AWS Schema Conversion Tool.

Pour plus d'informations sur l'utilisation AWS Secrets Manager, voir [Qu'est-ce que c'est AWS Secrets Manager ?](#) dans le guide de AWS Secrets Manager l'utilisateur. Pour plus d'informations sur le stockage AWS des profils, consultez [Gestion des profils dans le AWS Schema Conversion Tool](#).

Pour récupérer les informations d'identification de base de données depuis Secrets Manager

1. Démarrez le AWS Schema Conversion Tool et créez-en un nouveau.
2. Choisissez Ajouter une source ou Ajouter une cible pour ajouter une nouvelle base de données à votre projet.
3. Choisissez une plate-forme de base de données, puis cliquez sur Suivant.
4. Pour AWS Secret, choisissez le secret que vous souhaitez utiliser.
5. Choisissez Remplir. AWS SCT Renseigne ensuite toutes les valeurs dans la boîte de dialogue de connexion à la base de données.
6. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données.
7. Choisissez Connect pour vous connecter à votre base de données.

AWS SCT prend en charge les secrets dont la structure est la suivante.

```
{
  "username": "secret_user",
  "password": "secret_password",
  "engine": "oracle",
  "host": "secret_host.eu-west-1.compute.amazonaws.com",
  "port": "1521",
  "dbname": "ora_db"
}
```

Dans cette structure, les `password` valeurs `username` et sont obligatoires, tandis que toutes les autres valeurs sont facultatives. Assurez-vous que les valeurs que vous stockez dans Secrets Manager incluent toutes les informations d'identification de base de données.

## Stockage des mots de passe dans le AWS Schema Conversion Tool

Vous pouvez enregistrer un mot de passe de base de données ou un certificat SSL dans le AWS SCT cache. Pour stocker un mot de passe, choisissez Store Password (Stocker le mot de passe) lorsque vous créez une connexion.

Le mot de passe est chiffré à l'aide du jeton généré de façon aléatoire dans le fichier `seed.dat`. Il est ensuite stocké avec le nom d'utilisateur dans le fichier `cache`. Si vous perdez le fichier `seed.dat` ou qu'il est endommagé, le mot de passe de base de données risque d'être déchiffré de façon incorrecte. Dans ce cas, la connexion échoue.

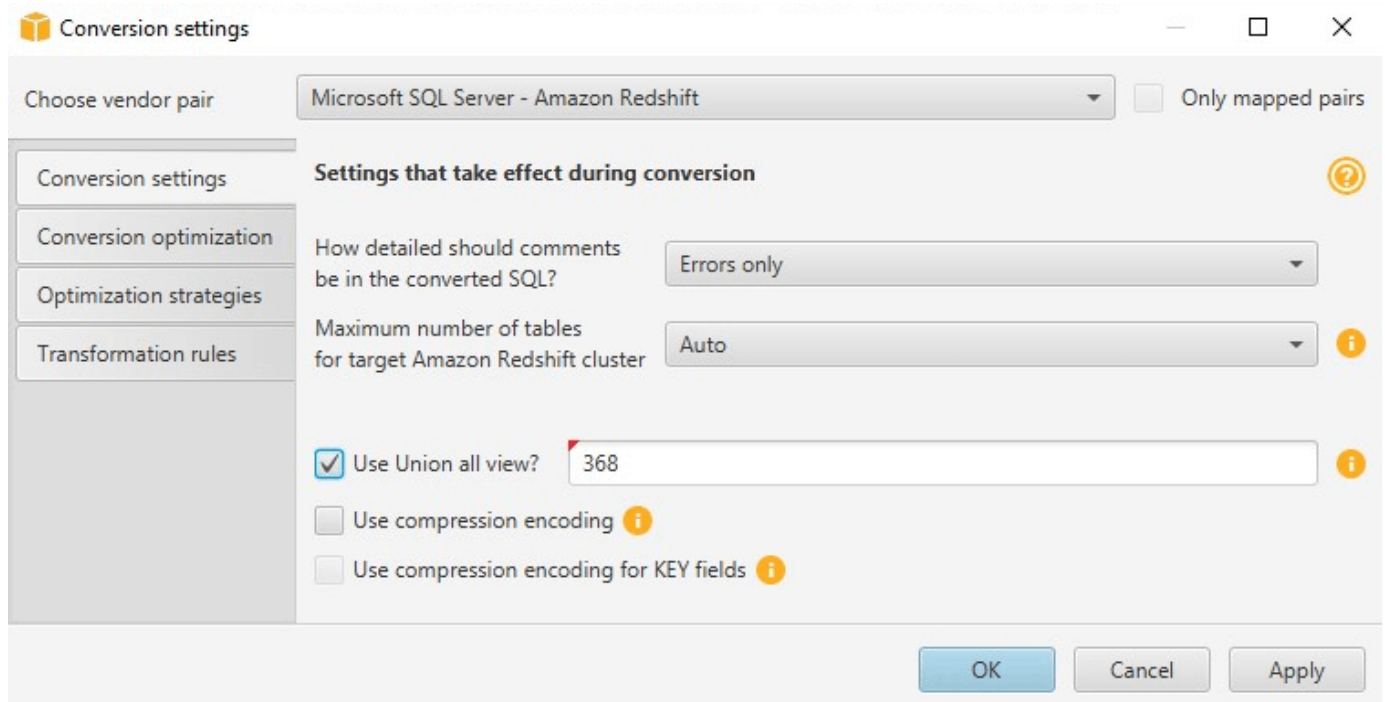
## Création de la vue UNION ALL dans le AWS Schema Conversion Tool

Si une table source est partitionnée, AWS SCT crée `n` tables cibles, où `n` est le nombre de partitions de la table source. AWS SCT crée une vue UNION ALL au-dessus des tables cibles pour représenter la table source. Si vous utilisez un extracteur de AWS SCT données pour migrer vos données, les partitions de la table source seront extraites et chargées en parallèle par des sous-tâches distinctes.

Pour utiliser la vue Union All pour un projet

1. Démarrer AWS SCT. Créez un nouveau projet ou ouvrez un AWS SCT projet existant.
2. Dans le menu Paramètres, choisissez Paramètres de conversion.

3. Choisissez une paire de bases de données OLAP dans la liste en haut.
4. Activer l'affichage complet d'Use Union ?



5. Cliquez sur OK pour enregistrer les paramètres et fermer la boîte de dialogue des paramètres de conversion.

## Utilisation des raccourcis clavier dans le AWS Schema Conversion Tool

Les raccourcis clavier que vous pouvez utiliser sont les suivants AWS SCT.

Raccourci clavier	Description
Ctrl+N	Crée un projet.
Ctrl+O	Ouvre un projet existant.
Ctrl+S	Enregistre un projet ouvert.
Ctrl+W	Crée un nouveau projet à l'aide de l'assistant.
Ctrl+M	Créez une nouvelle évaluation multiserveur.



Raccourci clavier	Description
Ctrl+L	Ajoutez une nouvelle base de données source.
Ctrl + R	Ajoutez une nouvelle base de données cible.
Ctrl+F4	Fermez un projet ouvert.
F1	Ouvrez le guide de AWS SCT l'utilisateur.

# Commencer avec AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir le schéma d'une base de données source. La base de données source peut être un moteur autogéré exécuté sur site ou sur une instance Amazon EC2 . Vous pouvez convertir votre schéma source en schéma pour toute base de données prise en charge hébergée par AWS. L' AWS SCT application fournit une interface utilisateur basée sur des projets.

Presque tout le travail que vous effectuez AWS SCT commence par les étapes suivantes :

1. Installez AWS SCT. Pour de plus amples informations, veuillez consulter [Installation et configuration AWS Schema Conversion Tool](#).
2. Installez un AWS SCT agent, si nécessaire. AWS SCT les agents ne sont nécessaires que pour certains scénarios de migration, par exemple entre des sources et des cibles hétérogènes. Pour de plus amples informations, veuillez consulter [Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool](#).
3. Familiarisez-vous avec l'interface utilisateur de AWS SCT. Pour de plus amples informations, veuillez consulter [Naviguer dans l'interface utilisateur du AWS SCT](#).
4. Créez un AWS SCT projet. Se connecter à vos bases de données source et cible. Pour plus d'informations sur la connexion à votre base de données source, consultez [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#).
5. Créez des règles de mappage. Pour plus d'informations sur les règles de mappage, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#).
6. Exécuter puis vérifier le rapport d'évaluation de migration de base de données. Pour plus d'informations sur le rapport d'évaluation, consultez [Afficher le rapport d'évaluation dans AWS Schema Conversion Tool](#).
7. Convertir les schémas de base de données source. Vous devez prendre en compte plusieurs aspects de la conversion, notamment concernant les actions à effectuer avec les éléments qui ne sont pas convertis et la façon de mapper les éléments qui doivent être convertis de façon particulière. Pour plus d'informations sur la conversion d'un schéma source, consultez [Conversion de schémas de base de données dans AWS Schema Conversion Tool](#).

Si vous convertissez un schéma d'entrepôt de données, vous devez également prendre en compte certains éléments avant de procéder à la conversion. Pour de plus amples informations, veuillez consulter [Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT](#).

8. Appliquez la conversion de schéma à votre cible. Pour plus d'informations sur l'application d'une conversion d'un schéma source, consultez [Appliquer des schémas convertis](#).
9. Vous pouvez également l'utiliser AWS SCT pour convertir des procédures stockées SQL et d'autres codes d'application. Pour plus d'informations, consultez [Conversion du code SQL d'une application en utilisant AWS SCT](#).

Vous pouvez également l'utiliser AWS SCT pour migrer vos données d'une base de données source vers une base de données gérée par Amazon. Pour obtenir des exemples, consultez [Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool](#).

# Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool

AWS Schema Conversion Tool (AWS SCT) peut convertir les schémas des bases de données sources et entrepôts de données suivants en une base de données cible ou en un entrepôt de données. Pour plus d'informations sur les autorisations, les connexions et les éléments AWS SCT pouvant être convertis pour une utilisation avec la base de données cible ou l'entrepôt de données, consultez les informations détaillées dans les rubriques suivantes.

## Informations de chiffrement

### [Connexion à Amazon RDS et/ou Aurora chiffrés](#)

## Sources de base de données

- [Connexion à Apache Cassandra](#)
- [Connexion à Azure SQL](#)
- [Connexion à IBM DB2 pour z/OS](#)
- [Bases de données IBM DB2 LUW](#)
- [Utilisation de MySQL comme source](#)
- [Bases de données Oracle](#)
- [Bases de données PostgreSQL](#)
- [Bases de données SAP](#)
- [Bases de données SQL Server](#)

## Sources de l'entrepôt de données

- [Amazon Redshift](#)
- [Azure Synapse Analytics en tant que source](#)
- [BigQuery comme source](#)
- [Bases de données Greenplum](#)
- [Bases de données Netezza](#)
- [Entrepôt de données Oracle](#)

- [Snowflake](#)
- [Entrepôts de données SQL Server](#)
- [Bases de données Teradata](#)
- [Bases de données Vertica](#)

#### Sources de données massives

- [Connexion à Apache Hadoop](#)
- [Connexion à Apache Oozie](#)

## Connexion aux bases de données chiffrées Amazon Relational Database Service et Amazon Aurora à l'aide du AWS Schema Conversion Tool

Pour ouvrir des connexions chiffrées aux bases de données Amazon RDS ou Amazon Aurora à partir d'une application, vous devez importer des certificats AWS racines dans une forme de stockage de clés. Vous pouvez télécharger les certificats racines AWS à l'adresse [Using SSL/TLS to encrypt a connection to a DB instance dans](#) le guide de l'utilisateur Amazon RDS.

Deux options sont disponibles : un certificat racine qui fonctionne pour toutes les AWS régions et un ensemble de certificats contenant à la fois l'ancien et le nouveau certificat racine.

En fonction de celui que vous souhaitez utiliser, suivez les étapes décrites dans l'une des deux procédures suivantes.


Pour importer le ou les certificats dans le stockage du système Windows

1. Téléchargez un ou plusieurs certificats à partir de l'une des sources suivantes :

Pour plus d'informations sur le téléchargement de certificats, consultez la section [Utilisation du protocole SSL/TLS pour chiffrer une connexion à une instance de base de données dans](#) le guide de l'utilisateur Amazon RDS.

2. Dans votre fenêtre de recherche Windows, entrez **Manage computer certificates**. Lorsque vous êtes invité à indiquer si vous souhaitez autoriser l'application à apporter des modifications à votre ordinateur, choisissez Oui.

3. Lorsque la fenêtre des certificats s'ouvre, si nécessaire, développez Certificats - Local Computer afin de voir la liste des certificats. Ouvrez le menu contextuel (clic droit) des autorités de certification Trusted Root, puis choisissez Toutes les tâches, Importer.
4. Choisissez Suivant, puis Parcourir, et recherchez le \*.pem fichier que vous avez téléchargé à l'étape 1. Choisissez Ouvrir pour sélectionner le fichier de certificat, cliquez sur Suivant, puis sur Terminer.

 Note

Pour trouver le fichier, remplacez le type de fichier dans la fenêtre de navigation par Tous les fichiers (\*.\*), car il ne \*.pem s'agit pas d'une extension de certificat standard.

5. Dans la console de gestion Microsoft, développez Certificats. Développez ensuite Trusted Root Certification Authorities, choisissez Certificates et recherchez le certificat pour confirmer son existence. Le nom du certificat commence par Amazon RDS.
6. Redémarrez votre ordinateur.

Pour importer le ou les certificats dans le Java KeyStore

1. Téléchargez le ou les certificats depuis l'une des sources suivantes :

Pour plus d'informations sur le téléchargement de certificats, consultez la section [Utilisation du protocole SSL/TLS pour chiffrer une connexion à une instance de base de données dans](#) le guide de l'utilisateur Amazon RDS.

2. Si vous avez téléchargé le bundle de certificats, divisez-le en fichiers de certificats individuels. Pour ce faire, placez chaque bloc de certificat, en commençant par -----BEGIN CERTIFICATE----- et en finissant par, -----END CERTIFICATE----- dans un \*.pem fichier distinct. Après avoir créé un \*.pem fichier distinct pour chaque certificat, vous pouvez supprimer le fichier du bundle de certificats en toute sécurité.
3. Ouvrez une fenêtre de commande ou une session de terminal dans le répertoire où vous avez téléchargé le certificat, puis exécutez la commande suivante pour chaque \*.pem fichier que vous avez créé à l'étape précédente.

```
keytool -importcert -file <filename>.pem -alias <filename>.pem -keystore storename
```

## Example

L'exemple suivant suppose que vous avez téléchargé le `eu-west-1-bundle.pem` fichier.

```
keytool -importcert -file eu-west-1-bundle.pem -alias eu-west-1-bundle.pem -
keystore trust-2019.ks
Picked up JAVA_TOOL_OPTIONS: -Dlog4j2.formatMsgNoLookups=true
Enter keystore password:
Re-enter new password:
Owner: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
  ST=Washington, L=Seattle, C=US
Issuer: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
  ST=Washington, L=Seattle, C=US
Serial number: c73467369250ae75
Valid from: Thu Aug 22 19:08:50 CEST 2019 until: Thu Aug 22 19:08:50 CEST 2024
Certificate fingerprints:
    SHA1: D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96
    SHA256:
    F2:54:C7:D5:E9:23:B5:B7:51:0C:D7:9E:F7:77:7C:1C:A7:E6:4A:3C:97:22:E4:0D:64:54:78:FC:70:AA:
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 73 5F 60 D8 BC CB 03 98   F4 2B 17 34 2E 36 5A A6   s_`.....+.4.6Z.
0010: 60 FF BC 1F                   `...
]
]

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
```

```

]

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 73 5F 60 D8 BC CB 03 98   F4 2B 17 34 2E 36 5A A6   s_`.....+.4.6Z.
0010: 60 FF BC 1F               `...
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

```

4. Ajoutez le keystore en tant que trust store. AWS SCT Pour ce faire, dans le menu principal, choisissez Paramètres, Paramètres globaux, Sécurité, Magasin de confiance, puis sélectionnez Sélectionner le magasin de confiance existant.

Après avoir ajouté le trust store, vous pouvez l'utiliser pour configurer une connexion SSL lorsque vous créez une AWS SCT connexion à la base de données. Dans la boîte de dialogue AWS SCT Connect to database, choisissez Use SSL et choisissez le trust store saisi précédemment.

## Connexion aux bases de données Apache Cassandra à l'aide du AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir les espaces clés d'Apache Cassandra en Amazon DynamoDB.

### Connexion à Apache Cassandra en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Apache Cassandra avec le AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Apache Cassandra

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Cassandra, puis Next.

La boîte de dialogue Ajouter une source apparaît.



3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source d'Apache Cassandra, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du service DNS (Domain Name Service) ou l'adresse IP du serveur de la base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le</p>

Paramètre	Action
	mot de passe pour vous connecter à votre base de données source selon vos besoins.
Use SSL	<p>Choisissez cette option si vous souhaitez utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Trust store : Le trust store à utiliser.</li> <li>• Magasin de clés : le magasin de clés à utiliser.</li> </ul>
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Connexion aux bases de données Apache Hadoop à l'aide du AWS Schema Conversion Tool

Vous pouvez utiliser l'interface de ligne de commande (CLI) de AWS SCT pour migrer d'Apache Hadoop vers Amazon EMR. AWS SCT utilise votre compartiment Amazon S3 comme espace de stockage temporaire pour vos données pendant la migration.

AWS SCT supporte en tant que source Apache Hadoop version 2.2.0 et supérieure. AWS SCT Supporte également Apache Hive version 0.13.0 et supérieure.

AWS SCT prend en charge comme cible les versions 6.3.0 et supérieures d'Amazon EMR. AWS SCT Supporte également en tant que cible Apache Hadoop version 2.6.0 et supérieure, et Apache Hive version 0.13.0 et supérieure.

### Rubriques

- [Conditions préalables à l'utilisation d'Apache Hadoop en tant que source](#)
- [Autorisations d'utilisation de Hive comme source](#)
- [Autorisations d'utilisation de HDFS comme source](#)
- [Autorisations d'utilisation de HDFS comme cible](#)
- [Connexion à Apache Hadoop en tant que source](#)
- [Connexion à vos services Hive et HDFS source](#)
- [Connexion à Amazon EMR en tant que cible](#)

## Conditions préalables à l'utilisation d'Apache Hadoop en tant que source

Les prérequis suivants sont requis pour se connecter à Apache Hadoop avec la CLI. AWS SCT

- Créez un compartiment Amazon S3 pour stocker les données pendant la migration. Vous pouvez ensuite copier les données vers Amazon EMR HDFS ou utiliser Amazon S3 comme référentiel de données pour vos charges de travail Hadoop. Pour plus d'informations, consultez la section [Création d'un compartiment](#) dans le guide de l'utilisateur Amazon S3.
- Créez un rôle AWS Identity and Access Management (IAM) avec la `AmazonS3FullAccess` politique. AWS SCT utilise ce rôle IAM pour accéder à votre compartiment Amazon S3.
- Prenez note de votre clé AWS secrète et de votre clé d'accès AWS secrète. Pour plus d'informations sur les clés AWS d'accès, consultez [la section Gestion des clés d'accès](#) dans le guide de l'utilisateur IAM.
- Créez et configurez un cluster Amazon EMR cible. Pour plus d'informations, consultez [Getting started with Amazon EMR](#) dans le guide de gestion Amazon EMR.
- Installez `dist-cputilitaire` sur votre cluster Apache Hadoop source. Installez également `s3-dist-cputilitaire` sur votre cluster Amazon EMR cible. Assurez-vous que les utilisateurs de votre base de données sont autorisés à exécuter ces utilitaires.
- Configurez le `core-site.xml` fichier dans votre cluster Hadoop source pour utiliser le protocole `s3a`. Pour ce faire, définissez le `fs.s3a.aws.credentials.provider` paramètre sur l'une des valeurs suivantes.
  - `org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider`
  - `org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider`
  - `org.apache.hadoop.fs.s3a.AnonymousAWSCredentialsProvider`
  - `org.apache.hadoop.fs.s3a.auth.AssumedRoleCredentialProvider`

Vous pouvez ajouter l'exemple de code suivant dans le `core-site.xml` fichier.

```
<property>
  <name>fs.s3a.aws.credentials.provider</name>
  <value>org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider</value>
</property>
```

L'exemple précédent montre l'une des quatre options de la liste d'options précédente. Si vous ne définissez pas le `fs.s3a.aws.credentials.provider` paramètre dans le `core-site.xml` fichier, AWS SCT choisit automatiquement le fournisseur.

## Autorisations d'utilisation de Hive comme source

Les autorisations requises pour un utilisateur source Hive sont les suivantes :

- READaccès aux dossiers de données source et au compartiment Amazon S3 source
- READ+WRITEaccès aux compartiments Amazon S3 intermédiaires et cibles

Pour accélérer la migration, nous vous recommandons d'exécuter le compactage des tables sources transactionnelles ACID.

Les autorisations requises pour un utilisateur cible d'Amazon EMR Hive sont les suivantes :

- READaccès au compartiment Amazon S3 cible
- READ+WRITEaccès au compartiment Amazon S3 intermédiaire
- READ+WRITEaccès aux dossiers HDFS cibles

## Autorisations d'utilisation de HDFS comme source

Les autorisations requises pour HDFS en tant que source sont les suivantes :

- EXECUTEpour le NameNode
- EXECUTE+READpour tous les dossiers et fichiers source que vous incluez dans le projet de migration
- READ+WRITEpour le tmp répertoire dans lequel NameNode exécuter les tâches Spark et stocker les fichiers avant la migration vers Amazon S3

Dans HDFS, toutes les opérations nécessitent un accès transversal. L'accès transversal exige l'EXECUTE autorisation sur tous les composants existants du chemin, à l'exception du dernier composant du chemin. Par exemple, pour toute opération d'accès/foο/bar/baz, votre utilisateur doit avoir EXECUTE l'autorisation sur //foο, et/foο/bar.

L'exemple de code suivant montre comment accorder EXECUTE+READ des autorisations pour vos dossiers et fichiers sources, ainsi que READ+WRITE des autorisations pour le tmp répertoire.

```
hadoop fs -chmod -R 744 /user/hdfs-data
hadoop fs -chmod -R 766 /tmp
```

## Autorisations d'utilisation de HDFS comme cible

Les autorisations requises pour Amazon EMR HDFS en tant que cible sont les suivantes :

- EXECUTE pour NameNode le cluster Amazon EMR cible
- READ+WRITE pour les dossiers HDFS cibles dans lesquels vous stockerez les données après la migration

## Connexion à Apache Hadoop en tant que source

Vous pouvez utiliser Apache Hadoop comme source dans la AWS SCT version 1.0.670 ou supérieure. Vous pouvez migrer des clusters Hadoop vers Amazon EMR uniquement via l'interface de AWS SCT ligne de commande (CLI). Avant de commencer, familiarisez-vous avec l'interface de ligne de commande de AWS SCT. Pour de plus amples informations, veuillez consulter [Référence CLI pour AWS Schema Conversion Tool](#).

Pour vous connecter à Apache Hadoop dans la CLI AWS SCT

1. Créez un nouveau script AWS SCT CLI ou modifiez un modèle de scénario existant. Par exemple, vous pouvez télécharger et modifier le HadoopMigrationTemplate.scts modèle. Pour de plus amples informations, veuillez consulter [Obtenir des scénarios CLI](#).
2. Configurez les paramètres de l' AWS SCT application tels que l'emplacement du pilote et le dossier journal.

Téléchargez le pilote JDBC requis et spécifiez l'emplacement où vous stockez le fichier. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

L'exemple de code suivant montre comment ajouter le chemin d'accès au pilote Apache Hive. Après avoir exécuté cet exemple de code, AWS SCT stocke les fichiers journaux dans le c:\sct dossier.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{
    "hive_driver_file": "c:\\sct\\HiveJDBC42.jar",
    "log_folder": "c:\\sct",
    "console_log_folder": "c:\\sct"
  }'
/
```

Vous pouvez utiliser cet exemple et les exemples suivants sous Windows.

### 3. Créez un nouveau AWS SCT projet.

L'exemple de code suivant crée le `hadoop_emr` projet dans le c:\sct dossier.

```
CreateProject
  -name: 'hadoop_emr'
  -directory: 'c:\sct'
/
```

### 4. Ajoutez votre cluster Hadoop source au projet.

Utilisez la `AddSourceCluster` commande pour vous connecter au cluster Hadoop source. Assurez-vous de fournir des valeurs pour les paramètres obligatoires suivants : `namehost`, `port`, et `user`. Les autres paramètres sont facultatifs.

L'exemple de code suivant ajoute le cluster Hadoop source. Cet exemple définit `HADOOP_SOURCE` comme nom le cluster source. Utilisez ce nom d'objet pour ajouter les services Hive et HDFS au projet et créer des règles de mappage.

```
AddSourceCluster
  -name: 'HADOOP_SOURCE'
  -vendor: 'HADOOP'
  -host: 'hadoop_address'
  -port: '22'
  -user: 'hadoop_user'
  -password: 'hadoop_password'
```

```
-useSSL: 'true'  
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: 'hadoop_passphrase'
```

/

Dans l'exemple précédent, remplacez *hadoop\_address* par l'adresse IP de votre cluster Hadoop. Si nécessaire, configurez la valeur de l'option de port. Ensuite, remplacez *hadoop\_user* et *hadoop\_password* par le nom de votre utilisateur Hadoop et le mot de passe de cet utilisateur. Pour *path\name*, entrez le nom et le chemin du fichier PEM de votre cluster Hadoop source.

5. Enregistrez votre script CLI. Ajoutez ensuite les informations de connexion pour vos services Hive et HDFS.

## Connexion à vos services Hive et HDFS source

Vous pouvez vous connecter à vos services Hive et HDFS sources à l'aide de la CLI AWS SCT . Pour vous connecter à Apache Hive, utilisez le pilote JDBC Hive version 2.3.4 ou supérieure. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

AWS SCT se connecte à Apache Hive avec l'utilisateur du hadoop cluster. Pour ce faire, utilisez les `AddSourceClusterHDFS` commandes `AddSourceClusterHive` et. Vous pouvez utiliser l'une des approches suivantes.

- Créez un nouveau tunnel SSH.

Pour `createTunnel`, saisissez **true**. Pour `host`, entrez l'adresse IP interne de votre service Hive ou HDFS source. Pour `port`, entrez le port de service de votre service Hive ou HDFS.

Entrez ensuite vos informations d'identification Hive ou HDFS pour `user` et `password` Pour plus d'informations sur les tunnels SSH, consultez [Configurer un tunnel SSH vers le nœud principal à l'aide de la redirection de port locale](#) dans le guide de gestion Amazon EMR.

- Utilisez un tunnel SSH existant.

Pour `host`, saisissez **localhost**. Pour `port`, entrez le port local à partir des paramètres du tunnel SSH.

- Connectez-vous directement à vos services Hive et HDFS.

Pour `host`, entrez l'adresse IP ou le nom d'hôte de votre service Hive ou HDFS source. Pour `port`, entrez le port de service de votre service Hive ou HDFS. Entrez ensuite vos informations d'identification Hive ou HDFS pour `user` et `password`

Pour vous connecter à Hive et à HDFS dans la CLI AWS SCT

1. Ouvrez votre script CLI qui inclut les informations de connexion pour votre cluster Hadoop source. Assurez-vous d'utiliser le nom du cluster Hadoop que vous avez défini à l'étape précédente.
2. Ajoutez votre service Hive source au projet.

Utilisez la `AddSourceClusterHive` commande pour connecter le service Hive source. Assurez-vous de fournir des valeurs pour les paramètres obligatoires suivants : `userpassword`, `cluster`, `name`, et `port`. Les autres paramètres sont facultatifs.

L'exemple de code suivant crée un tunnel AWS SCT pour travailler avec votre service Hive. Ce service Hive source fonctionne sur le même PC que AWS SCT. Cet exemple utilise le cluster `HADOOP_SOURCE` source de l'exemple précédent.

```
AddSourceClusterHive
  -cluster: 'HADOOP_SOURCE'
  -name: 'HIVE_SOURCE'
  -host: 'localhost'
  -port: '10005'
  -user: 'hive_user'
  -password: 'hive_password'
  -createTunnel: 'true'
  -localPort: '10005'
  -remoteHost: 'hive_remote_address'
  -remotePort: 'hive_port'
```

/

L'exemple de code suivant permet de se connecter à votre service Hive sans tunnel.

```
AddSourceClusterHive
  -cluster: 'HADOOP_SOURCE'
  -name: 'HIVE_SOURCE'
  -host: 'hive_address'
  -port: 'hive_port'
```



```
-user: 'hive_user'  
-password: 'hive_password'  
/
```

Dans les exemples précédents, remplacez *hive\_user* et *hive\_password* par le nom de votre utilisateur Hive et le mot de passe de cet utilisateur.

Ensuite, remplacez *hive\_address* et *hive\_port* par l'adresse NameNode IP et le port de votre cluster Hadoop source.

En *hive\_remote\_address* effet, vous pouvez utiliser la valeur par défaut 127.0.0.1 ou l'adresse NameNode IP de votre service Hive source.

### 3. Ajoutez votre service HDFS source au projet.

Utilisez la `AddSourceClusterHDFS` commande pour connecter le service HDFS source. Assurez-vous de fournir des valeurs pour les paramètres obligatoires suivants : `userpassword`, `cluster`, `name`, et `port`. Les autres paramètres sont facultatifs.

Assurez-vous que votre utilisateur dispose des autorisations requises pour migrer les données depuis votre service HDFS source. Pour de plus amples informations, veuillez consulter [Autorisations d'utilisation de Hive comme source](#).

L'exemple de code suivant crée un tunnel AWS SCT pour fonctionner avec votre service Apache HDFS. Cet exemple utilise le cluster `HADOOP_SOURCE` source que vous avez créé précédemment.

```
AddSourceClusterHDFS  
-cluster: 'HADOOP_SOURCE'  
-name: 'HDFS_SOURCE'  
-host: 'localhost'  
-port: '9005'  
-user: 'hdfs_user'  
-password: 'hdfs_password'  
-createTunnel: 'true'  
-localPort: '9005'  
-remoteHost: 'hdfs_remote_address'  
-remotePort: 'hdfs_port'  
/
```

Le code suivant permet de se connecter à votre service Apache HDFS sans tunnel.

```
AddSourceClusterHDFS
  -cluster: 'HADOOP_SOURCE'
  -name: 'HDFS_SOURCE'
  -host: 'hdfs_address'
  -port: 'hdfs_port'
  -user: 'hdfs_user'
  -password: 'hdfs_password'
/
```

Dans les exemples précédents, remplacez *hdfs\_user* et *hdfs\_password* par le nom de votre utilisateur HDFS et le mot de passe de cet utilisateur.

Ensuite, remplacez *hdfs\_address* et *hdfs\_port* par l'adresse NameNode IP et le port de votre cluster Hadoop source.

En *hdfs\_remote\_address* effet, vous pouvez utiliser la valeur par défaut 127.0.0.1 ou l'adresse NameNode IP de votre service Hive source.

4. Enregistrez votre script CLI. Ajoutez ensuite les informations de connexion pour votre cluster Amazon EMR cible, ainsi que les commandes de migration.

## Connexion à Amazon EMR en tant que cible

Vous pouvez vous connecter à votre cluster Amazon EMR cible à l'aide de la CLI AWS SCT . Pour ce faire, vous autorisez le trafic entrant et utilisez SSH. Dans ce cas, AWS SCT possède toutes les autorisations requises pour travailler avec votre cluster Amazon EMR. Pour plus d'informations, consultez [Before you connect](#) and [connect to the primary node using SSH](#) dans le guide de gestion Amazon EMR.

AWS SCT se connecte à Amazon EMR Hive avec l'utilisateur du hadoop cluster. Pour vous connecter à Amazon EMR Hive, utilisez le pilote Hive JDBC version 2.6.2.1002 ou ultérieure. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

Pour vous connecter à Amazon EMR dans la CLI AWS SCT

1. Ouvrez votre script CLI qui inclut les informations de connexion pour votre cluster Hadoop source. Ajoutez les informations d'identification Amazon EMR cibles dans ce fichier.
2. Ajoutez votre cluster Amazon EMR cible au projet.

L'exemple de code suivant ajoute le cluster Amazon EMR cible. Cet exemple définit HADOOP\_TARGET comme nom le cluster cible. Utilisez ce nom d'objet pour ajouter vos services Hive et HDFS ainsi qu'un dossier de compartiment Amazon S3 au projet et créer des règles de mappage.

```
AddTargetCluster
  -name: 'HADOOP_TARGET'
  -vendor: 'AMAZON_EMR'
  -host: 'ec2-44-44-55-66.eu-west-1.EXAMPLE.amazonaws.com'
  -port: '22'
  -user: 'emr_user'
  -password: 'emr_password'
  -useSSL: 'true'
  -privateKeyPath: 'c:\path\name.pem'
  -passPhrase: '1234567890abcdef0!'
  -s3Name: 'S3_TARGET'
  -accessKey: 'AKIAIOSFODNN7EXAMPLE'
  -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
  -region: 'eu-west-1'
  -s3Path: 'doc-example-bucket/example-folder'
/
```

Dans l'exemple précédent, entrez les noms de vos AWS ressources et les informations de connexion Amazon EMR. Cela inclut l'adresse IP de votre cluster Amazon EMR, la clé AWS d'accès, la clé d'accès AWS secrète et le compartiment Amazon S3. Si nécessaire, configurez la valeur de la variable de port. Remplacez ensuite *emr\_user* et *emr\_password* par le nom de votre utilisateur Amazon EMR et le mot de passe de cet utilisateur. Pour *path\name*, entrez le nom et le chemin d'accès au fichier PEM de votre cluster Amazon EMR cible. Pour plus d'informations, voir [Télécharger le fichier PEM pour accéder au cluster EMR](#).

### 3. Ajoutez votre compartiment Amazon S3 cible au projet.

L'exemple de code suivant ajoute le compartiment Amazon S3 cible. Cet exemple utilise le HADOOP\_TARGET cluster que vous avez créé précédemment.

```
AddTargetClusterS3
  -cluster: 'HADOOP_TARGET'
  -Name: 'S3_TARGET'
  -accessKey: 'AKIAIOSFODNN7EXAMPLE'
  -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
```

```
-region: 'eu-west-1'  
-s3Path: 'doc-example-bucket/example-folder'  
/  

```

Dans l'exemple précédent, entrez votre clé AWS d'accès, votre clé d'accès AWS secrète et votre compartiment Amazon S3.

#### 4. Ajoutez votre service Hive cible au projet.

L'exemple de code suivant crée un tunnel AWS SCT pour travailler avec votre service Hive cible. Cet exemple utilise le cluster HADOOP\_TARGET cible que vous avez créé précédemment.

```
AddTargetClusterHive  
-cluster: 'HADOOP_TARGET'  
-name: 'HIVE_TARGET'  
-host: 'localhost'  
-port: '10006'  
-user: 'hive_user'  
-password: 'hive_password'  
-createTunnel: 'true'  
-localPort: '10006'  
-remoteHost: 'hive_address'  
-remotePort: 'hive_port'  
/  

```

Dans l'exemple précédent, remplacez *hive\_user* et *hive\_password* par le nom de votre utilisateur Hive et le mot de passe de cet utilisateur.

Remplacez ensuite *hive\_address* par la valeur par défaut 127.0.0.1 ou par l'adresse NameNode IP de votre service Hive cible. Remplacez-le ensuite *hive\_port* par le port de votre service Hive cible.

#### 5. Ajoutez votre service HDFS cible au projet.

L'exemple de code suivant crée un tunnel AWS SCT pour fonctionner avec votre service Apache HDFS. Cet exemple utilise le cluster HADOOP\_TARGET cible que vous avez créé précédemment.

```
AddTargetClusterHDFS  
-cluster: 'HADOOP_TARGET'  
-name: 'HDFS_TARGET'  
-host: 'localhost'  
-port: '8025'  

```

```
-user: 'hdfs_user'  
-password: 'hdfs_password'  
-createTunnel: 'true'  
-localPort: '8025'  
-remoteHost: 'hdfs_address'  
-remotePort: 'hdfs_port'
```

/

Dans l'exemple précédent, remplacez *hdfs\_user* et *hdfs\_password* par le nom de votre utilisateur HDFS et le mot de passe de cet utilisateur.

Ensuite, remplacez *hdfs\_address* et *hdfs\_port* par l'adresse IP privée et le port NameNode de votre service HDFS cible.

6. Enregistrez votre script CLI. Ajoutez ensuite des règles de mappage et des commandes de migration. Pour de plus amples informations, veuillez consulter [Migration des charges de travail Hadoop](#).

## Connexion aux flux de travail Apache Oozie à l'aide du AWS Schema Conversion Tool

Vous pouvez utiliser l'interface de ligne de commande (CLI) de AWS SCT pour convertir les flux de travail Apache Oozie en AWS Step Functions. Après avoir migré vos charges de travail Apache Hadoop vers Amazon EMR, vous pouvez utiliser un service natif pour orchestrer vos tâches. AWS Cloud Pour de plus amples informations, veuillez consulter [Connexion à Apache Hadoop](#).

AWS SCT convertit vos flux de travail Oozie en AWS Step Functions et les utilise pour AWS Lambda émuler des fonctionnalités que AWS Step Functions ne supporte pas. AWS SCT Convertit également les propriétés de votre tâche Oozie en AWS Systems Manager

Pour convertir les flux de travail Apache Oozie, assurez-vous d'utiliser la version 1.0.671 de AWS SCT ou supérieure. Familiarisez-vous également avec l'interface de ligne de commande de AWS SCT. Pour de plus amples informations, veuillez consulter [Référence CLI pour AWS Schema Conversion Tool](#).

### Conditions préalables à l'utilisation d'Apache Oozie en tant que source

Les prérequis suivants sont requis pour se connecter à Apache Oozie avec la CLI. AWS SCT

- Créez un compartiment Amazon S3 pour stocker les définitions des machines d'état. Vous pouvez utiliser ces définitions pour configurer vos machines d'état. Pour plus d'informations, consultez la section [Création d'un compartiment](#) dans le guide de l'utilisateur Amazon S3.
- Créez un rôle AWS Identity and Access Management (IAM) avec la `AmazonS3FullAccess` politique. AWS SCT utilise ce rôle IAM pour accéder à votre compartiment Amazon S3.
- Prenez note de votre clé AWS secrète et de votre clé d'accès AWS secrète. Pour plus d'informations sur les clés AWS d'accès, consultez [la section Gestion des clés d'accès](#) dans le guide de l'utilisateur IAM.
- Stockez vos AWS informations d'identification et les informations relatives à votre compartiment Amazon S3 dans le profil de AWS service des paramètres globaux de l'application. Ensuite, AWS SCT utilise ce profil de AWS service pour utiliser vos AWS ressources. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).

Pour travailler avec vos flux de travail Apache Oozie source, vous devez AWS SCT disposer de la structure spécifique de vos fichiers source. Chacun de vos dossiers de candidature doit inclure le `job.properties` fichier. Ce fichier inclut les paires clé-valeur des propriétés de votre tâche. En outre, chacun de vos dossiers de candidature doit inclure le `workflow.xml` fichier. Ce fichier décrit les nœuds d'action et les nœuds de flux de contrôle de votre flux de travail.

## Connexion à Apache Oozie en tant que source

Utilisez la procédure suivante pour vous connecter à vos fichiers source Apache Oozie.

Pour vous connecter à Apache Oozie dans la CLI AWS SCT

1. Créez un nouveau script AWS SCT CLI ou modifiez un modèle de scénario existant. Par exemple, vous pouvez télécharger et modifier le `OozieConversionTemplate.scts` modèle. Pour de plus amples informations, veuillez consulter [Obtenir des scénarios CLI](#).
2. Configurez les paramètres de AWS SCT l'application.

L'exemple de code suivant enregistre les paramètres de l'application et permet de stocker des mots de passe dans votre projet. Vous pouvez utiliser ces paramètres enregistrés dans d'autres projets.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{
```

```
        "store_password": "true"  
    }'  
/'
```

### 3. Créez un nouveau AWS SCT projet.

L'exemple de code suivant crée le `oozie` projet dans le `c:\sct` dossier.

```
CreateProject  
  -name: 'oozie'  
  -directory: 'c:\sct'  
/'
```

### 4. Ajoutez le dossier contenant vos fichiers source Apache Oozie au projet à l'aide de la `AddSource` commande. Assurez-vous d'utiliser la `APACHE_00ZIE` valeur du `vendor` paramètre. Fournissez également des valeurs pour les paramètres obligatoires suivants : `name` et `mappingsFolder`.

L'exemple de code suivant ajoute Apache Oozie en tant que source dans votre AWS SCT projet. Cet exemple crée un objet source portant le nom `00ZIE`. Utilisez ce nom d'objet pour ajouter des règles de mappage. Après avoir exécuté cet exemple de code, AWS SCT utilise le `c:\oozie` dossier pour charger vos fichiers source dans le projet.

```
AddSource  
  -name: '00ZIE'  
  -vendor: 'APACHE_00ZIE'  
  -mappingsFolder: 'c:\oozie'  
/'
```

Vous pouvez utiliser cet exemple et les exemples suivants sous Windows.

### 5. Connectez-vous à vos fichiers source Apache Oozie à l'aide de la `ConnectSource` commande. Utilisez le nom de votre objet source que vous avez défini à l'étape précédente.

```
ConnectSource  
  -name: '00ZIE'  
  -mappingsFolder: 'c:\oozie'  
/'
```

### 6. Enregistrez votre script CLI. Ajoutez ensuite les informations de connexion pour votre AWS Step Functions service.

## Autorisations d'utilisation AWS Lambda des fonctions du pack d'extension

Pour les fonctions source qui AWS Step Functions ne sont pas prises en charge, AWS SCT crée un pack d'extension. Ce pack d'extension inclut AWS Lambda des fonctions qui émulent vos fonctions sources.

Pour utiliser ce pack d'extension, créez un rôle AWS Identity and Access Management (IAM) avec les autorisations suivantes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "lambda",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:498160209112:function:LoadParameterInitialState:*",
        "arn:aws:lambda:*:498160209112:function:EvaluateJSPELExpressions:*"
      ]
    },
    {
      "Sid": "emr",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeStep",
        "elasticmapreduce:AddJobFlowSteps"
      ],
      "Resource": [
        "arn:aws:elasticmapreduce:*:498160209112:cluster/*"
      ]
    },
    {
      "Sid": "s3",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/*"
      ]
    }
  ]
}
```



```

    }
  ]
}

```

Pour appliquer le pack d'extension, vous AWS SCT devez disposer d'un rôle IAM doté des autorisations suivantes.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:CreateRole",
        "iam:TagRole",
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy",
        "iam>DeleteRole",
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::ACCOUNT_NUMBER:role/sct/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies"
      ],
      "Resource": [
        "arn:aws:iam::ACCOUNT_NUMBER:role/lambda_LoadParameterInitialStateRole",
        "arn:aws:iam::ACCOUNT_NUMBER:role/lambda_EvaluateJSPELExpressionsRole",
        "arn:aws:iam::ACCOUNT_NUMBER:role/stepFunctions_MigratedOozieWorkflowRole"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [

```

```
        "lambda:GetFunction",
        "lambda:CreateFunction",
        "lambda:UpdateFunctionCode",
        "lambda>DeleteFunction"
    ],
    "Resource": [
        "arn:aws:lambda:*:ACCOUNT_NUMBER:function:LoadParameterInitialState",
        "arn:aws:lambda:*:ACCOUNT_NUMBER:function:EvaluateJSPELExpressions"
    ]
}
]
```

## Connexion à AWS Step Functions en tant que cible

Pour vous connecter en tant que cible, procédez AWS Step Functions comme suit.

Pour vous connecter AWS Step Functions à la AWS SCT CLI

1. Ouvrez votre script CLI qui inclut les informations de connexion pour vos fichiers source Apache Oozie.
2. Ajoutez les informations relatives à votre cible de migration dans le AWS SCT projet à l'aide de la `AddTarget` commande. Assurez-vous d'utiliser la `STEP_FUNCTIONS` valeur du `vendor` paramètre. Fournissez également des valeurs pour les paramètres obligatoires suivants : `name` et `profile`.

L'exemple de code suivant est ajouté en AWS Step Functions tant que source dans votre AWS SCT projet. Cet exemple crée un objet cible portant le nom `AWS_STEP_FUNCTIONS`. Utilisez ce nom d'objet lorsque vous créez des règles de mappage. Cet exemple utilise également un profil de AWS SCT service que vous avez créé à l'étape des conditions préalables. Assurez-vous de le remplacer *profile\_name* par le nom de votre profil.

```
AddTarget
  -name: 'AWS_STEP_FUNCTIONS'
  -vendor: 'STEP_FUNCTIONS'
  -profile: 'profile_name'
/
```

Si vous n'utilisez pas le profil de AWS service, assurez-vous de fournir des valeurs pour les paramètres obligatoires suivants : `accessKeysecretKey`, `awsRegion`, et `s3Path`. Utilisez ces

paramètres pour spécifier votre clé d'accès AWS AWS secrète, votre clé secrète et le chemin d'accès à votre compartiment Amazon S3. Région AWS

3. Connectez-vous à AWS Step Functions l'aide de la `ConnectTarget` commande. Utilisez le nom de l'objet cible que vous avez défini à l'étape précédente.

L'exemple de code suivant permet de se connecter à l'objet `AWS_STEP_FUNCTIONS` cible à l'aide de votre profil de AWS service. Assurez-vous de le remplacer `profile_name` par le nom de votre profil.

```
ConnectTarget
  -name: 'AWS_STEP_FUNCTIONS'
  -profile: 'profile_name'
/
```

4. Enregistrez votre script CLI. Ajoutez ensuite des règles de mappage et des commandes de migration. Pour de plus amples informations, veuillez consulter [Conversion des flux de travail Oozie](#) ;.

## Connexion aux bases de données Microsoft Azure SQL à l'aide du AWS SCT

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application depuis Azure SQL Database vers les cibles suivantes :

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

### Rubriques

- [Privilèges pour Azure SQL Database en tant que source](#)
- [Connexion à Azure SQL Database en tant que source](#)

## Privilèges pour Azure SQL Database en tant que source

Les privilèges requis pour Azure SQL Database en tant que source sont les suivants :

- VIEW DEFINITION
- VIEW DATABASE STATE

Répétez l'octroi pour chaque base de données dont vous convertissez le schéma.

Les privilèges requis pour les bases de données MySQL et PostgreSQL cibles sont décrits dans les sections suivantes.

- [Privilèges pour MySQL en tant que base de données cible](#)
- [Privilèges pour PostgreSQL en tant que base de données cible](#)

## Connexion à Azure SQL Database en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Azure SQL Database avec le AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Azure SQL Database

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Azure SQL Database, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Azure SQL Database, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du service DNS (Domain Name Service) ou l'adresse IP du serveur de la base de données source.
Database (Base de données)	Entrez le nom de la base de données à laquelle vous souhaitez vous connecter.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.

5. Choisissez **Tester la connexion** pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez **Connect** pour vous connecter à votre base de données source.

# Connexion aux bases de données IBM DB2 for z/OS à l'aide du AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application d'IBM Db2 for z/OS vers les cibles suivantes.

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

## Conditions requises pour utiliser DB2 pour z/OS en tant que base de données source

La version de base de données IBM Db2 for z/OS version 12, niveau de fonction 100, ne prend pas en charge la plupart des nouvelles fonctionnalités d'IBM Db2 for z/OS version 12. Cette version de base de données prend en charge le retour à la version 11 de DB2 et le partage de données avec DB2 version 11. Pour éviter la conversion de fonctionnalités non prises en charge par la version 11 de DB2, nous vous recommandons d'utiliser une fonction de base de données IBM Db2 for z/OS de niveau 500 ou supérieur comme source pour AWS SCT.

Vous pouvez utiliser l'exemple de code suivant pour vérifier la version de votre base de données source IBM Db2 for z/OS.

```
SELECT GETVARIABLE('SYSIBM.VERSION') as version FROM SYSIBM.SYSDUMMY1;
```

Assurez-vous que ce code renvoie la version DSN12015 ou une version supérieure.

Vous pouvez utiliser l'exemple de code suivant pour vérifier la valeur du registre APPLICATION COMPATIBILITY spécial dans votre base de données source IBM Db2 for z/OS.

```
SELECT CURRENT APPLICATION COMPATIBILITY as version FROM SYSIBM.SYSDUMMY1;
```

Assurez-vous que ce code renvoie la version V12R1M500 ou une version supérieure.

## Privilèges pour Db2 for z/OS en tant que base de données source

Les privilèges nécessaires pour se connecter à une base de données DB2 pour z/OS et lire les catalogues et les tables du système sont les suivants :

- SÉLECTIONNEZ SUR SYSIBM.LOCATIONS
- SÉLECTIONNEZ SUR SYSIBM.SYSCHECKS
- SÉLECTIONNEZ SUR SYSIBM.SYSCOLUMNS
- SÉLECTIONNEZ SUR SYSIBM.SYSDATABASE
- SÉLECTIONNEZ SUR SYSIBM.SYSDATATYPES
- SÉLECTIONNEZ SUR SYSIBM.SYSDUMMY1
- SÉLECTIONNEZ SUR SYSIBM.SYSFOREIGNKEYS
- SÉLECTIONNEZ SUR SYSIBM.SYSINDEXES
- SÉLECTIONNEZ SUR SYSIBM.SYSKEYCOLUSE
- SÉLECTIONNEZ SUR SYSIBM.SYSKEYS
- SÉLECTIONNEZ SUR SYSIBM.SYSKEYTARGETS
- SÉLECTIONNEZ SUR SYSIBM.SYSJAROBJECTS
- SÉLECTIONNEZ SUR SYSIBM.SYSPACKAGE
- SÉLECTIONNEZ SUR SYSIBM.SYSPARMS
- SÉLECTIONNEZ SUR SYSIBM.SYSRELS
- SÉLECTIONNEZ SUR SYSIBM.SYSROUTINES
- SÉLECTIONNEZ SUR SYSIBM.SYSSEQUENCES
- SÉLECTIONNEZ SUR SYSIBM.SYSSEQUENCESDEP
- SÉLECTIONNEZ SUR SYSIBM.SYSSYNONYMES
- SÉLECTIONNEZ SUR SYSIBM.SYSTABCONST
- SÉLECTIONNEZ SUR SYSIBM.SYSTABLES
- SÉLECTIONNEZ SUR SYSIBM.SYSTABLESPACE
- SÉLECTIONNEZ SUR SYSIBM.SYSTRIGGERS
- SÉLECTIONNEZ SUR SYSIBM.SYSVARIABLES
- SÉLECTIONNEZ SUR SYSIBM.SYSVIEWS

Pour convertir des tables DB2 pour z/OS en tables partitionnées PostgreSQL, collectez des statistiques sur les tablespaces et les tables de votre base de données à l'aide de l'utilitaire comme indiqué ci-dessous. RUNSTATS

```
LISTDEF YOURLIST INCLUDE TABLESPACES DATABASE YOURDB
RUNSTATS TABLESPACE
LIST YOURLIST
TABLE (ALL) INDEX (ALL KEYCARD)
UPDATE ALL
REPORT YES
SHRLEVEL REFERENCE
```

Dans l'exemple précédent, remplacez l'*YOURDB* espace réservé par le nom de la base de données source.

## Connexion à DB2 pour z/OS en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source DB2 pour z/OS avec AWS SCT

Pour se connecter à une base de données source IBM Db2 for z/OS

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Db2 pour z/OS, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.



Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source IBM Db2 for z/OS, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Emplacement	Entrez le nom unique de l'emplacement DB2 auquel vous souhaitez accéder.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option si vous souhaitez utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>Trust store : emplacement d'un trust store contenant des certificats. Pour que cet emplacement apparaisse ici, assurez-vous de l'ajouter dans les paramètres généraux.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Chemin du pilote DB2 pour z/OS	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

- Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
- Choisissez Connect pour vous connecter à votre base de données source.

## Privilèges pour MySQL en tant que base de données cible

Les privilèges requis pour MySQL en tant que cible sont les suivants :

- CRÉER SUR\* . \*

- MODIFIER \*.\*
- DÉPOSEZ \*.\*
- INDEX SUR \*.\*
- RÉFÉRENCES SUR\*.\*
- SELECT ON \*.\*
- CRÉER UNE VUE SUR \*.\*
- SHOW VIEW ON \*.\*
- DÉCLENCHEUR ACTIVÉ\*.\*
- CRÉER UNE ROUTINE SUR\*.\*
- MODIFIER LA ROUTINE SUR \*.\*
- EXÉCUTER SUR\*.\*
- SELECT ON mysql.proc
- INSÉRER, METTRE À JOUR SUR AWS\_DB 2ZOS\_EXT.\*
- INSÉREZ, METTEZ À JOUR, SUPPRIMEZ SUR AWS\_DB 2ZOS\_EXT\_DATA.\*
- CRÉEZ DES TABLES TEMPORAIRES SUR AWS\_DB 2ZOS\_EXT\_DATA.\*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT SELECT ON mysql.proc TO 'user_name';
GRANT INSERT, UPDATE ON AWS_DB2ZOS_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_DB2ZOS_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_DB2ZOS_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for MySQL comme cible, définissez `log_bin_trust_function_creators` le paramètre sur `true`, puis `character_set_server` sur `latin1` to. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

Pour utiliser Aurora MySQL comme cible, définissez le `log_bin_trust_function_creators` paramètre sur `true` et sur `character_set_server` `tolatin1`. Définissez également le `lower_case_table_names` paramètre sur `true`. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

## Privilèges pour PostgreSQL en tant que base de données cible

Pour utiliser PostgreSQL comme cible AWS SCT , le privilège est requis. `CREATE ON DATABASE` Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser Amazon RDS for PostgreSQL comme cible, le privilège est AWS SCT requis.  
`rds_superuser`

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par `"$user"`, `public_synonyms`, `public`.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';
GRANT CREATE ON DATABASE db_name TO user_name;
GRANT rds_superuser TO user_name;
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Dans PostgreSQL, seul le propriétaire du schéma ou un `superuser` peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

## Paramètres de conversion de DB2 pour z/OS vers PostgreSQL

Pour modifier les paramètres de conversion de DB2 pour z/OS vers PostgreSQL, choisissez Paramètres, puis Paramètres de conversion. Dans la liste supérieure, choisissez Db2 pour z/OS, puis choisissez Db2 pour z/OS — PostgreSQL ou Db2 pour z/OS — Amazon Aurora (compatible avec PostgreSQL). AWS SCT affiche tous les paramètres disponibles pour la conversion d'IBM Db2 for z/OS vers PostgreSQL.

Les paramètres de conversion de DB2 pour z/OS vers PostgreSQL incluent des options pour les éléments suivants : AWS SCT

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour générer des noms uniques pour les contraintes dans la base de données cible.

Dans PostgreSQL, tous les noms de contrainte que vous utilisez doivent être uniques. AWS SCT peut générer des noms uniques pour les contraintes dans le code converti en ajoutant un préfixe avec le nom de la table au nom de votre contrainte. Pour vous assurer que cela AWS SCT génère des noms uniques pour vos contraintes, sélectionnez Générer des noms uniques pour les contraintes.

- Pour conserver le formatage des noms de colonnes, des expressions et des clauses dans les instructions DML du code converti.

AWS SCT permet de conserver la disposition des noms de colonnes, des expressions et des clauses dans les instructions DML dans la même position et dans le même ordre que dans le code source. Pour ce faire, sélectionnez Oui pour conserver le formatage des noms de colonnes, des expressions et des clauses dans les instructions DML.

- Pour exclure les partitions de table de la zone de conversion.

AWS SCT peut ignorer toutes les partitions d'une table source lors de la conversion. Pour ce faire, sélectionnez Exclure les partitions de table de la zone de conversion.

- Utiliser le partitionnement automatique pour les tables partitionnées en fonction de la croissance.

Pour la migration des données, AWS SCT vous pouvez partitionner automatiquement toutes les tables dont la taille est supérieure à la taille spécifiée. Pour utiliser cette option, sélectionnez Appliquer la partition des tables dont la taille est supérieure à, puis entrez la taille des tables en gigaoctets. Entrez ensuite le nombre de partitions. AWS SCT prend en compte la taille du périphérique de stockage à accès direct (DASD) de votre base de données source lorsque vous activez cette option.

AWS SCT peut déterminer le nombre de partitions automatiquement. Pour ce faire, sélectionnez Augmenter le nombre de partitions proportionnellement et entrez le nombre maximum de partitions.

- Pour renvoyer des ensembles de résultats dynamiques sous forme de tableau de valeurs du type de données refcursor.

AWS SCT peut convertir les procédures source qui renvoient des ensembles de résultats dynamiques en procédures comportant un ensemble de curseurs ouverts comme paramètre de sortie supplémentaire. Pour ce faire, sélectionnez Utiliser un tableau de curseurs pour renvoyer tous les ensembles de résultats dynamiques.

- Spécifier la norme à utiliser pour la conversion des valeurs de date et d'heure en représentations sous forme de chaînes.

AWS SCT peut convertir les valeurs de date et d'heure en représentations sous forme de chaînes en utilisant l'un des formats industriels pris en charge. Pour ce faire, sélectionnez Utiliser des représentations sous forme de chaînes de valeurs de date ou Utiliser des représentations sous forme de chaînes de valeurs temporelles. Choisissez ensuite l'une des normes suivantes.

- Organisation internationale de normalisation (ISO)
- Norme européenne IBM (EUR)
- IBM USA Standard (États-Unis)
- Norme industrielle japonaise de l'ère chrétienne (JIS)

# Connexion aux bases de données IBM DB2 pour Linux, UNIX et Windows avec le AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code dans le langage SQL et du code d'application d'IBM Db2 pour Linux, Unix et Windows (Db2 LUW) vers les cibles suivantes.

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for MariaDB

AWS SCT prend en charge comme source les versions 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 et 11.5 de Db2 LUW.

## Privilèges pour DB2 LUW en tant que source

Les privilèges nécessaires pour se connecter à une base de données DB2 LUW, vérifier les privilèges disponibles et lire les métadonnées du schéma d'une source sont les suivants :

- Privilège requis pour établir une connexion :
  - CONNEXION À LA BASE DE DONNÉES
- Privilège requis pour exécuter les instructions SQL :
  - EXÉCUTER SUR LE PACKAGE NULLID. SYSSH200
- Privilèges requis pour obtenir les informations au niveau de l'instance :
  - EXÉCUTER SUR LA FONCTION SYSPROC.ENV\_GET\_INST\_INFO
  - SÉLECTIONNEZ SUR SYSIBMADM.ENV\_INST\_INFO
  - SÉLECTIONNEZ SUR SYSIBMADM.ENV\_SYS\_INFO
- Privilèges requis pour vérifier les privilèges accordés par le biais de rôles, de groupes et d'autorités :
  - EXÉCUTER SUR LA FONCTION SYSPROC.AUTH\_LIST\_AUTHORTIES\_FOR\_AUTHID
  - EXÉCUTER SUR LA FONCTION SYSPROC.AUTH\_LIST\_GROUPS\_FOR\_AUTHID
  - EXÉCUTER SUR LA FONCTION SYSPROC.AUTH\_LIST\_ROLES\_FOR\_AUTHID

- SÉLECTIONNEZ SUR SYSIBMADM.PRIVILEGES
- Privilèges requis sur les catalogues et tables système :
  - SÉLECTIONNEZ SUR SYSCAT.ATTRIBUTES
  - SÉLECTIONNEZ SUR SYSCAT.CHECKS
  - SÉLECTIONNEZ SUR SYSCAT.COLIDENTATTRIBUTES
  - SÉLECTIONNEZ SUR SYSCAT.COLUMNNS
  - SÉLECTIONNEZ SUR SYSCAT.DATAPARTITIONEXPRESSION
  - SÉLECTIONNEZ SUR SYSCAT.DATAPARTITIONS
  - SÉLECTIONNEZ SUR SYSCAT.DATATYPEDEP
  - SÉLECTIONNEZ SUR SYSCAT.DATATYPES
  - SÉLECTIONNEZ SUR SYSCAT.HIERARCHIES
  - SÉLECTIONNEZ SUR SYSCAT.INDEXCOLUSE
  - SÉLECTIONNEZ SUR SYSCAT.INDEXES
  - SÉLECTIONNEZ SUR SYSCAT.INDEXPARTITIONS
  - SÉLECTIONNEZ SUR SYSCAT.KEYCOLUSE
  - SÉLECTIONNER SUR SYSCAT.MODULEOBJECTS
  - SÉLECTIONNEZ SUR SYSCAT.MODULES
  - SÉLECTIONNEZ SUR SYSCAT.NICKNAMES
  - SÉLECTIONNEZ SUR SYSCAT.PERIODS
  - SÉLECTIONNEZ SUR SYSCAT.REFERENCES
  - SÉLECTIONNEZ SUR SYSCAT.ROUTINEPARMS
  - SÉLECTIONNEZ SUR SYSCAT.ROUTINES
  - SÉLECTIONNEZ SUR SYSCAT.ROWFIELDS
  - SÉLECTIONNEZ SUR SYSCAT.SCHEMATA
  - SÉLECTIONNEZ SUR SYSCAT.SEQUENCES
  - SÉLECTIONNEZ SUR SYSCAT.TABCONST
  - SÉLECTIONNEZ SUR SYSCAT.TABLES
  - SÉLECTIONNEZ SUR SYSCAT.TRIGGERS
  - SÉLECTIONNEZ SUR SYSCAT.VARIABLEDEP
- SÉLECTIONNEZ SUR SYSCAT.VARIABLES



- SÉLECTIONNEZ SUR SYSCAT.VIEWS
- SÉLECTIONNEZ SUR SYSIBM. SYSDUMMY1
- Pour exécuter des instructions SQL, le compte utilisateur a besoin d'un privilège pour utiliser au moins l'une des charges de travail activées dans la base de données. Si aucune des charges de travail n'est affectée à l'utilisateur, vérifiez que la charge de travail utilisateur par défaut est accessible par l'utilisateur :
  - UTILISATION SUR LA CHARGE DE TRAVAIL SYSDEFAULTUSERWORKLOAD

Pour exécuter des requêtes, vous devez créer des espaces de table temporaires pour le système avec une taille de page de 8 k, 16 k et 32 k, si elles n'existent pas déjà. Pour créer les espaces de table temporaires, exécutez les scripts suivants.

```
CREATE BUFFERPOOL BP8K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 8K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_8K
  PAGESIZE 8192
  BUFFERPOOL BP8K;

CREATE BUFFERPOOL BP16K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 16K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP16K
  PAGESIZE 16384
  BUFFERPOOL BP16K;

CREATE BUFFERPOOL BP32K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 32K;
```

```
CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP32K
PAGE SIZE 32768
BUFFERPOOL BP32K;
```

## Connexion à DB2 LUW en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Db2 LUW avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Db2 LUW

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Db2 LUW, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source IBM Db2 LUW, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.

Paramètre	Action
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Entrez le nom de la base de données Db2 LUW.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>
Use SSL	<p>Choisissez cette option si vous souhaitez utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Trust store : emplacement d'un trust store contenant des certificats. Pour que cet emplacement apparaisse ici, assurez-vous de l'ajouter dans les paramètres généraux.</li></ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>

Paramètre	Action
Chemin du pilote DB2 LUW	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Migration d'IBM DB2 pour Linux, UNIX et Windows vers Amazon Relational Database Service pour PostgreSQL ou Amazon Aurora PostgreSQL compatible Edition

Lorsque vous migrez IBM Db2 LUW vers PostgreSQL, vous AWS SCT pouvez convertir diverses instructions de déclenchement utilisées avec Db2 LUW. Il s'agit notamment des instructions de déclenchement suivantes :

- Événements déclencheurs : les événements déclencheurs INSERT, DELETE et UPDATE spécifient que l'action déclenchée s'exécute chaque fois que l'événement est appliqué à la table ou à la vue du sujet. Vous pouvez définir n'importe quelle combinaison des événements INSERT, DELETE et UPDATE, mais vous ne pouvez spécifier chaque événement qu'une seule fois. AWS SCT prend en charge les événements déclencheurs uniques et multiples. Concernant les événements, PostgreSQL possède quasiment les mêmes fonctionnalités.
- ÉVÉNEMENT DE COLONNE — Vous pouvez spécifier un nom de colonne à partir d'une table de base. Le déclencheur est activé uniquement par la mise à jour d'une colonne qui est identifiée dans la colonne des noms de liste. PostgreSQL possède les mêmes fonctionnalités.
- Déclencheurs d'instructions : ils spécifient que l'action déclenchée n'est appliquée qu'une seule fois pour l'ensemble de l'instruction. Vous ne pouvez pas spécifier ce type de granularité de

déclencheur pour un déclencheur BEFORE ou un déclencheur INSTEAD OF. Si cette valeur est spécifiée, un déclencheur UPDATE ou DELETE est activé, même si aucune ligne n'est affectée. PostgreSQL possède également cette fonctionnalité et la déclaration de déclenchement pour les déclencheurs d'instruction est identique pour PostgreSQL et Db2 LUW.

- **Clauses de référencement** : elles spécifient les noms de corrélation pour les variables de transition et les noms de table pour les tables de transition. Les noms de corrélation identifient une ligne spécifique dans l'ensemble des lignes affectées par l'opération SQL de déclenchement. Les noms de table identifient l'ensemble complet des lignes affectées. Chaque ligne affectée par une opération SQL de déclenchement est disponible pour l'action déclenchée en identifiant les colonnes avec les noms de corrélation spécifiés. PostgreSQL ne prend pas en charge cette fonctionnalité, et utilise uniquement un nom de corrélation NOUVEAU ou ANCIEN.
- **AU LIEU DE DÉCLENCHEURS**, il AWS SCT les prend en charge.

## Conversion de tables partitionnées DB2 LUW en tables partitionnées PostgreSQL version 10

AWS SCT peut convertir des tables DB2 LUW en tables partitionnées dans PostgreSQL 10. Il existe plusieurs restrictions lors de la conversion d'une table partitionnée Db2 LUW en PostgreSQL :

- Vous pouvez créer une table partitionnée avec une colonne acceptant la valeur null dans Db2 LUW, et vous pouvez spécifier une partition pour stocker les valeurs NULL. Cependant, PostgreSQL ne prend pas en charge les valeurs NULL pour le partitionnement RANGE.
- Db2 LUW peut utiliser une clause INCLUSIVE ou EXCLUSIVE pour définir des valeurs de limite de plage. PostgreSQL prend uniquement en charge INCLUSIVE pour une limite de début et EXCLUSIVE pour une limite de fin. Le nom de partition converti est au format <nom\_table\_origine>\_<nom\_partition\_origine>.
- Vous pouvez créer des clés primaires ou uniques pour les tables partitionnées dans Db2 LUW. Pour PostgreSQL, vous devez une clé primaire ou unique pour chaque partition directement. Les contraintes de clé primaire ou unique doivent être supprimées de la table parent. Le nom de clé converti est au format <nom\_clé\_origine>\_<nom\_partition\_origine>.
- Vous pouvez créer une contrainte de clé étrangère depuis et vers une table partitionnée dans Db2 LUW. Cependant, PostgreSQL ne prend pas en charge les références de clé étrangère dans les tables partitionnées. PostgreSQL ne prend pas non plus en charge les références de clé étrangère à partir d'une table partitionnée vers une autre table.

- Vous pouvez créer un index sur une table partitionnée dans Db2 LUW. Cependant, pour PostgreSQL, vous devez un index pour chaque partition directement. Les index doivent être supprimés de la table parent. Le nom d'index converti est au format `<nom_index_origine>_<nom_partition_origine>`.
- Vous devez définir des déclencheurs de ligne sur les partitions individuelles, et non sur la table partitionnée. Les déclencheurs doivent être supprimés de la table parent. Le nom de déclencheur converti est au format `<nom_déclencheur_origine>_<nom_partition_origine>`.

## Privilèges pour PostgreSQL en tant que cible

Pour utiliser PostgreSQL comme cible AWS SCT , le privilège est requis. `CREATE ON DATABASE` Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par `"$user", public_synonyms, public`.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Dans PostgreSQL, seul le propriétaire du schéma ou un `superuser` peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

# Migration d'IBM DB2 pour Linux, UNIX et Windows vers Amazon RDS for MySQL ou Amazon Aurora MySQL

Lorsque vous convertissez une base de données IBM Db2 LUW en RDS for MySQL ou Amazon Aurora MySQL, tenez compte de ce qui suit.

## Privilèges pour MySQL en tant que cible

Les privilèges requis pour MySQL en tant que cible sont les suivants :

- CRÉER SUR\* . \*
- MODIFIER \* . \*
- DÉPOSEZ \* . \*
- INDEX SUR \* . \*
- RÉFÉRENCES SUR\* . \*
- SELECT ON \*.\*
- CRÉER UNE VUE SUR \* . \*
- SHOW VIEW ON \*.\*
- DÉCLENCHEUR ACTIVÉ\* . \*
- CRÉER UNE ROUTINE SUR\* . \*
- MODIFIER LA ROUTINE SUR \* . \*
- EXÉCUTER SUR\* . \*
- SELECT ON mysql.proc
- INSÉRER, METTRE À JOUR SUR AWS\_DB 2\_EXT. \*
- INSÉREZ, METTEZ À JOUR, SUPPRIMEZ SUR AWS\_DB 2\_EXT\_DATA. \*
- CRÉEZ DES TABLES TEMPORAIRES SUR AWS\_DB 2\_EXT\_DATA. \*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';  
GRANT CREATE ON *.* TO 'user_name';  
GRANT ALTER ON *.* TO 'user_name';
```

```
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT SELECT ON mysql.proc TO 'user_name';
GRANT INSERT, UPDATE ON AWS_DB2_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_DB2_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_DB2_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques DETERMINISTIC, READS SQL DATA ni NO SQL pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

## Utiliser MySQL comme source pour AWS SCT

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code de base de données et du code d'application depuis MySQL vers les cibles suivantes :

- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for MySQL

Pour plus d'informations, consultez les sections suivantes :

### Rubriques



- [Privilèges pour MySQL en tant que base de données source](#)
- [Connexion à MySQL en tant que source](#)
- [Privilèges pour PostgreSQL en tant que base de données cible](#)

## Privilèges pour MySQL en tant que base de données source

Les privilèges requis pour MySQL en tant que source sont les suivants :

- SELECT ON \*.\*
- SHOW VIEW ON \*.\*

## Connexion à MySQL en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source MySQL avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source MySQL

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez MySQL, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :

- Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
  1. Pour AWS Secret, choisissez le nom du secret.
  2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source MySQL, suivez les instructions suivantes :

Paramètre	Action
Server name	<p>Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.</p> <p>Vous pouvez vous connecter à votre base de données MySQL source à l'aide d'un protocole d'IPv6 adresse. Pour ce faire, assurez-vous d'utiliser des crochets pour saisir l'adresse IP, comme indiqué dans l'exemple suivant.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans un projet. Pour éviter d'exposer le mot de passe de votre base de données source, AWS SCT ne le stocke pas par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Exiger le protocole SSL : choisissez cette option pour vous connecter au serveur uniquement via SSL.</li></ul> <p>Si vous choisissez Require SSL et que le serveur ne prend pas en charge SSL, vous ne pouvez pas vous connecter au serveur. Si vous ne choisissez pas Require SSL et que le serveur ne prend pas en charge SSL, vous pouvez quand même vous connecter au serveur sans utiliser SSL. Pour plus d'informations, consultez <a href="#">Configuration de MySQL pour utiliser des connexions sécurisées</a>.</p> <ul style="list-style-type: none"><li>• Vérifier le certificat du serveur : sélectionnez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li><li>• Trust store : emplacement d'un trust store contenant des certificats.</li></ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. L'activation de cette option vous permet de stocker le mot de passe de la base de données et de vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>

Paramètre	Action
MySql trajectoire du conducteur	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Privilèges pour PostgreSQL en tant que base de données cible

Pour utiliser PostgreSQL comme cible AWS SCT, le privilège est requis. CREATE ON DATABASE Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par "\$user", public\_synonyms, public.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';
GRANT CREATE ON DATABASE db_name TO user_name;
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Dans PostgreSQL, seul le propriétaire du schéma ou un superuser peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

## Connexion aux bases de données Oracle à l'aide du AWS Schema Conversion Tool

Vous pouvez les utiliser AWS SCT pour convertir des schémas, des objets de code de base de données et du code d'application depuis Oracle Database vers les cibles suivantes :

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for Oracle
- Amazon RDS for MariaDB

Lorsque la source est une base de données Oracle, les commentaires peuvent être convertis au format approprié, par exemple dans une base de données PostgreSQL. AWS SCT peut convertir les commentaires sur les tables, les vues et les colonnes. Les commentaires peuvent inclure des apostrophes ; les apostrophes sont AWS SCT doublées lors de la conversion d'instructions SQL, comme c'est le cas pour les chaînes littérales.

Pour plus d'informations, consultez les rubriques suivantes.

### Rubriques

- [Privilèges pour Oracle en tant que source](#)
- [Connexion à Oracle en tant que source](#)
- [Migration d'Oracle vers Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL avec AWS Schema Conversion Tool](#)
- [Migration d'Oracle vers Amazon RDS for MySQL ou Amazon Aurora MySQL avec AWS Schema Conversion Tool](#)
- [Migration d'une base de données Oracle vers Amazon RDS for Oracle avec AWS Schema Conversion Tool](#)

## Privilèges pour Oracle en tant que source

Les privilèges requis pour Oracle en tant que source sont les suivants :

- CONNECT
- SELECT\_CATALOG\_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT\$

## Connexion à Oracle en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Oracle avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Oracle

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Oracle, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :

- Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
  1. Pour AWS Secret, choisissez le nom du secret.
  2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Oracle, suivez les instructions suivantes :

Paramètre	Action
Type	<p>Choisissez le type de connexion à la base de données. Selon le type choisi, fournissez les informations complémentaires suivantes :</p> <ul style="list-style-type: none"><li>• SID<ul style="list-style-type: none"><li>• Nom du serveur : nom DNS (Domain Name System) ou adresse IP de votre serveur de base de données source.</li><li>• Server port : Port utilisé pour vous connecter au serveur de la base de données source.</li><li>• Oracle SID : ID du système Oracle (SID). Pour trouver le SID Oracle, soumettez la requête suivante à votre base de données Oracle : <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre></li></ul></li><li>• Nom du service<ul style="list-style-type: none"><li>• Server name : Nom DNS ou adresse IP du serveur de votre base de données source.  Vous pouvez vous connecter à votre base de données Oracle source à l'aide d'un protocole d'IPv6 adresse. Pour ce faire, assurez-vous d'utiliser des crochets pour saisir l'adresse IP, comme indiqué dans l'exemple suivant. <div data-bbox="716 1444 1507 1528" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div></li><li>• Server port : Port utilisé pour vous connecter au serveur de la base de données source.</li><li>• Nom du service : nom du service Oracle auquel se connecter.</li></ul></li><li>• Alias TNS</li></ul>

Paramètre	Action
	<ul style="list-style-type: none"><li>• TNS file path : Chemin d'accès au fichier qui contient les informations de connexion de nom TNS (Transparent Network Substrate).</li></ul> <p>Après avoir choisi le fichier TNS, il AWS SCT ajoute toutes les connexions à la base de données Oracle depuis le fichier à la liste des alias TNS.</p> <p>Choisissez cette option pour vous connecter à Oracle Real Application Clusters (RAC).</p> <ul style="list-style-type: none"><li>• Alias TNS : alias TNS de ce fichier à utiliser pour se connecter à la base de données source.</li><li>• Identifiant de connexion TNS</li><li>• Identifiant de connexion TNS : identifiant des informations de connexion TNS enregistrées.</li></ul>



Paramètre	Action
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>La première fois que vous vous connectez à la base de données Oracle, vous entrez le chemin d'accès au fichier du pilote Oracle (ojdbc8.jar). Vous pouvez télécharger le fichier à l'<a href="http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html">adresse http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html</a>. Assurez-vous de vous inscrire sur le site Web gratuit d'Oracle Technical Network pour terminer le téléchargement. AWS SCT utilise le pilote sélectionné pour toute future connexion à la base de données Oracle. Le chemin du pilote peut être modifié à l'aide de l'onglet Pilotes dans les paramètres généraux.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans un projet. Pour éviter d'exposer le mot de passe de votre base de données source, AWS SCT ne le stocke pas par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Authentification SSL : sélectionnez cette option pour utiliser l'authentification SSL par certificat Configurez votre magasin de confiance et votre magasin de clés dans Paramètres, Paramètres globaux, Sécurité.</li> <li>• Trust store : Le trust store à utiliser.</li> <li>• Magasin de clés : le magasin de clés à utiliser.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. Choisissez cette option pour enregistrer le mot de passe de la base de données et pour vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Chemin du pilote Oracle	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous enregistrez le chemin du pilote dans les paramètres globaux du projet, le chemin du pilote n'apparaît pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

# Migration d'Oracle vers Amazon RDS for PostgreSQL ou Amazon Aurora PostgreSQL avec AWS Schema Conversion Tool

Lorsque vous convertissez une base de données Oracle en RDS pour PostgreSQL ou Amazon Aurora PostgreSQL, tenez compte des points suivants.

## Rubriques

- [Privilèges pour PostgreSQL en tant que base de données cible](#)
- [Paramètres de conversion Oracle vers PostgreSQL](#)
- [Conversion de séquences Oracle](#)
- [Conversion de Oracle ROWID](#)
- [Conversion du SQL dynamique Oracle](#)
- [Conversion de partitions Oracle](#)

Lorsque vous convertissez des objets système Oracle en PostgreSQL AWS SCT , effectuez les conversions comme indiqué dans le tableau suivant.

Objet du système Oracle	Description	Objet PostgreSQL converti
V\$VERSION	Affiche les numéros de version des éléments de la bibliothèque principale dans la base de données Oracle	aws_oracle_ext.v\$version
V\$INSTANCE	Vue qui indique l'état de l'instance actuelle.	aws_oracle_ext.v\$instance

Vous pouvez l'utiliser AWS SCT pour convertir des fichiers Oracle SQL\*Plus en psql, une interface basée sur un terminal pour PostgreSQL. Pour de plus amples informations, veuillez consulter [Conversion du code SQL d'une application en utilisant AWS SCT](#).

## Privilèges pour PostgreSQL en tant que base de données cible

Pour utiliser PostgreSQL comme cible AWS SCT , le privilège est requis. `CREATE ON DATABASE` Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par "\$user", public\_synonyms, public.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for PostgreSQL comme cible, le privilège est AWS SCT requis.  
rds\_superuser

Dans PostgreSQL, seul le propriétaire du schéma ou un superuser peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle superuser.

## Paramètres de conversion Oracle vers PostgreSQL

Pour modifier les paramètres de conversion d'Oracle vers PostgreSQL, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Oracle, puis Oracle — PostgreSQL. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Oracle vers PostgreSQL.

Les paramètres de conversion d'Oracle vers PostgreSQL incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Permettre AWS SCT de convertir les vues matérialisées Oracle en tables ou en vues matérialisées sur PostgreSQL. Pour la conversion des vues matérialisées en tant que, choisissez le mode de conversion de vos vues matérialisées source.
- Pour travailler avec votre code source Oracle lorsqu'il inclut les TO\_NUMBER fonctions TO\_CHAR TO\_DATE, et avec des paramètres non pris en charge par PostgreSQL. Par défaut, AWS SCT émule l'utilisation de ces paramètres dans le code converti.

Lorsque votre code source Oracle inclut uniquement des paramètres pris en charge par PostgreSQL, vous pouvez utiliser PostgreSQL TO\_CHAR natif et ses fonctions. TO\_DATE TO\_NUMBER Dans ce cas, le code converti fonctionne plus rapidement. Pour inclure uniquement ces paramètres, sélectionnez les valeurs suivantes :

- La fonction TO\_CHAR () n'utilise pas de chaînes de formatage spécifiques à Oracle
- La fonction TO\_DATE () n'utilise pas de chaînes de formatage spécifiques à Oracle
- La fonction TO\_NUMBER () n'utilise pas de chaînes de formatage spécifiques à Oracle
- Pour remédier au cas où votre base de données source Oracle ne stocke que des valeurs entières dans les colonnes de clé principale ou étrangère du type de NUMBER données, AWS SCT vous pouvez convertir ces colonnes en type de BIGINT données. Cette approche améliore les performances de votre code converti. Pour adopter cette approche, sélectionnez Convertir les colonnes de clé principale/étrangère NUMBER en colonnes BIGINT. Assurez-vous que votre source n'inclut pas de valeurs à virgule flottante dans ces colonnes pour éviter toute perte de données.
- Pour ignorer les déclencheurs et les contraintes désactivés dans votre code source. Pour ce faire, choisissez Ignorer les déclencheurs et les contraintes désactivés.
- À utiliser AWS SCT pour convertir des variables de chaîne appelées en tant que SQL dynamique. Le code de la base de données peut modifier les valeurs de ces variables de chaîne. Pour vous assurer que cela convertit AWS SCT toujours la dernière valeur de cette variable de chaîne, sélectionnez Convertir le code SQL dynamique créé dans des routines appelées.
- Pour remédier à ce problème, les versions 10 et antérieures de PostgreSQL ne prennent pas en charge les procédures. Si vous ou vos utilisateurs ne connaissez pas l'utilisation des procédures dans PostgreSQL AWS SCT , vous pouvez convertir les procédures Oracle en fonctions PostgreSQL. Pour ce faire, sélectionnez Convertir les procédures en fonctions.

- Pour obtenir des informations supplémentaires sur les actions effectuées. Pour ce faire, vous pouvez ajouter des fonctions spécifiques au pack d'extension en sélectionnant Ajouter un bloc de levée d'exception pour les problèmes de migration présentant les niveaux de gravité suivants. Choisissez ensuite les niveaux de gravité pour déclencher les exceptions définies par l'utilisateur.
- Pour travailler avec une base de données Oracle source susceptible d'inclure des contraintes liées aux noms générés automatiquement. Si votre code source utilise ces noms, assurez-vous de sélectionner Convertir les noms de contraintes générés par le système à l'aide des noms d'origine de la source. Si votre code source utilise ces contraintes mais pas leur nom, désactivez cette option pour augmenter la vitesse de conversion.
- Pour déterminer si votre base de données et vos applications s'exécutent dans des fuseaux horaires différents. Par défaut, AWS SCT émule les fuseaux horaires dans le code converti. Toutefois, vous n'avez pas besoin de cette émulation lorsque votre base de données et vos applications utilisent le même fuseau horaire. Dans ce cas, sélectionnez Le fuseau horaire du côté client correspond au fuseau horaire du serveur.
- Pour déterminer si vos bases de données source et cible s'exécutent dans des fuseaux horaires différents. Dans ce cas, la fonction qui émule la fonction Oracle SYSDATE intégrée renvoie des valeurs différentes de celles de la fonction source. Pour vous assurer que vos fonctions source et cible renvoient les mêmes valeurs, choisissez Définir le fuseau horaire par défaut pour l'émulation SYSDATE.
- Pour utiliser les fonctions de l'extension Oracle dans votre code converti. Pour ce faire, dans Utiliser l'implémentation d'Oracle, sélectionnez les fonctions à utiliser. Pour plus d'informations sur Oracle, voir [Oracle](#) on GitHub.

## Conversion de séquences Oracle

AWS SCT convertit des séquences d'Oracle vers PostgreSQL. Si vous utilisez des séquences pour maintenir des contraintes d'intégrité, assurez-vous que les nouvelles valeurs d'une séquence migrée ne chevauchent pas les valeurs existantes.

Pour renseigner les séquences converties avec la dernière valeur de la base de données source

1. Ouvrez votre AWS SCT projet avec Oracle comme source.
2. Choisissez Paramètres, puis Paramètres de conversion.
3. Dans la liste supérieure, choisissez Oracle, puis Oracle — PostgreSQL. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Oracle vers PostgreSQL.
4. Choisissez Remplir les séquences converties avec la dernière valeur générée côté source.

5. Cliquez sur OK pour enregistrer les paramètres et fermer la boîte de dialogue des paramètres de conversion.

## Conversion de Oracle ROWID

Dans une base de données Oracle, la pseudo-colonne ROWID contient l'adresse de la ligne de table. La pseudocolonne ROWID est propre à Oracle. Elle est donc AWS SCT convertie en colonne de données dans PostgreSQL. En utilisant cette conversion, vous pouvez conserver les informations ROWID.

Lors de la conversion de la pseudocolonne ROWID, AWS SCT vous pouvez créer une colonne de données avec le type de données `bigint`. S'il n'existe aucune clé primaire, AWS SCT définit la colonne ROWID comme clé primaire. Si une clé primaire existe, AWS SCT définit la colonne ROWID avec une contrainte unique.

Si le code de votre base de données source inclut des opérations avec ROWID, que vous ne pouvez pas exécuter à l'aide d'un type de données numérique, vous AWS SCT pouvez créer une colonne de données avec ce type de `character varying` données.

Pour créer une colonne de données pour Oracle ROWID pour un projet

1. Ouvrez votre AWS SCT projet avec Oracle comme source.
2. Choisissez Paramètres, puis Paramètres de conversion.
3. Dans la liste supérieure, choisissez Oracle, puis Oracle — PostgreSQL. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Oracle vers PostgreSQL.
4. Pour Générer un identifiant de ligne, effectuez l'une des opérations suivantes :
  - Choisissez Générer comme identité pour créer une colonne de données numériques.
  - Choisissez Générer comme type de domaine de caractères pour créer une colonne de données de caractères.
5. Cliquez sur OK pour enregistrer les paramètres et fermer la boîte de dialogue des paramètres de conversion.

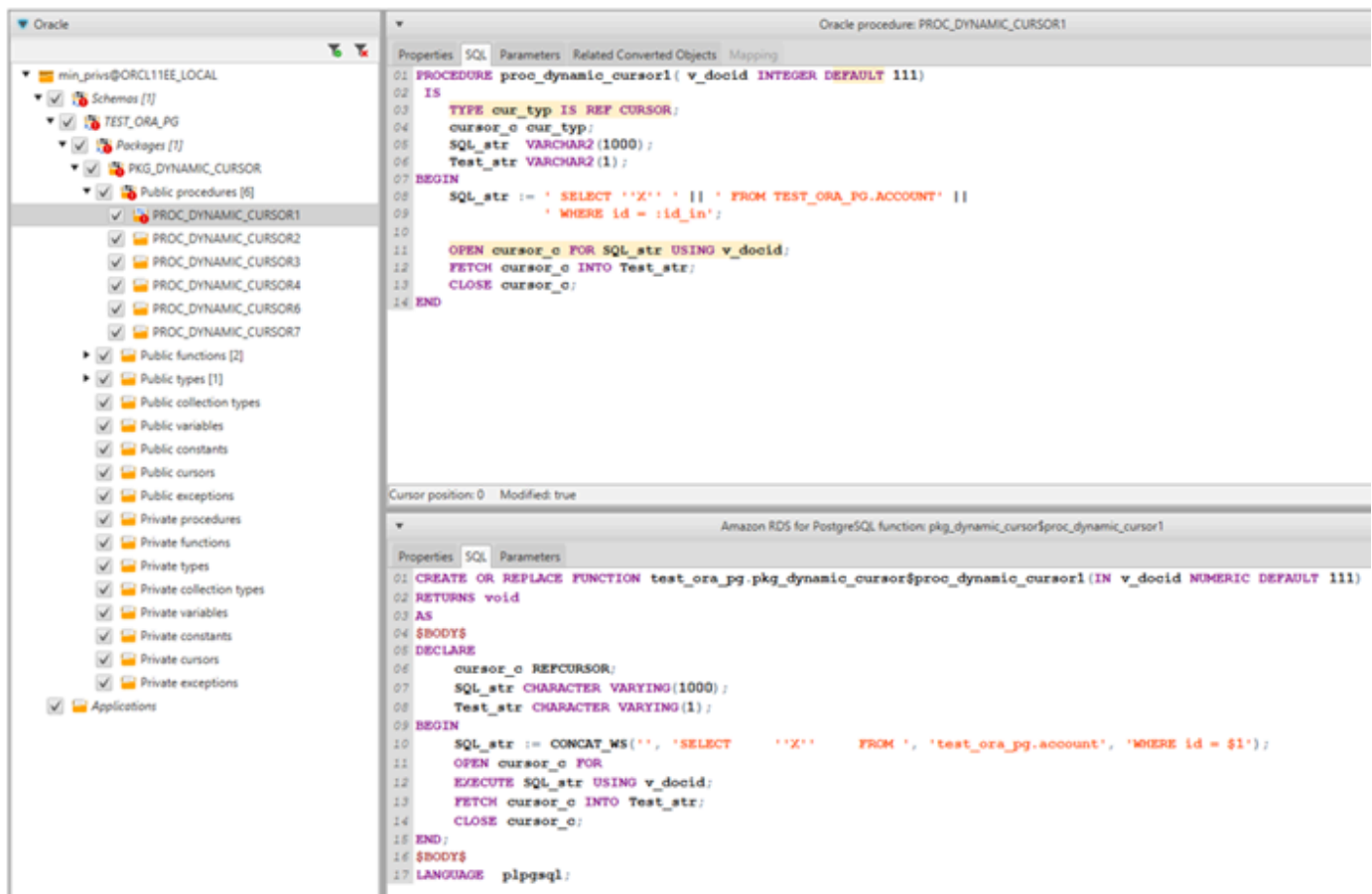
## Conversion du SQL dynamique Oracle

Oracle propose deux méthodes pour implémenter le SQL dynamique : en utilisant une instruction `EXECUTE IMMEDIATE` ou en appelant des procédures dans le package `DBMS_SQL`. Si votre base

de données Oracle source inclut des objets avec du SQL dynamique, utilisez-les AWS SCT pour convertir les instructions Oracle Dynamic SQL en PostgreSQL.

Pour convertir le SQL dynamique Oracle en PostgreSQL

1. Ouvrez votre AWS SCT projet avec Oracle comme source.
2. Choisissez un objet de base de données qui utilise le SQL dynamique dans l'arborescence des sources Oracle.
3. Ouvrez le menu contextuel (clic droit) de l'objet, choisissez Convertir le schéma et acceptez de remplacer les objets s'ils existent. La capture d'écran suivante montre la procédure convertie sous la procédure Oracle avec du SQL dynamique.



## Conversion de partitions Oracle

AWS SCT prend actuellement en charge les méthodes de partitionnement suivantes :

- Range
- Liste



- Gamme multicolonne
- Hachage
- Composite (liste de liste, liste de plages, plage de listes, hachage de liste, hachage de plage, hachage de hachage)

## Migration d'Oracle vers Amazon RDS for MySQL ou Amazon Aurora MySQL avec AWS Schema Conversion Tool

Pour émuler les fonctions de base de données Oracle dans votre code MySQL converti, utilisez le pack d'extension Oracle to MySQL dans AWS SCT. Pour plus d'informations sur les packs d'extension, consultez [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

### Rubriques

- [Privilèges pour MySQL en tant que base de données cible](#)
- [Paramètres de conversion Oracle vers MySQL](#)
- [Considérations concernant la migration](#)
- [Conversion de l'instruction WITH dans Oracle en RDS pour MySQL ou Amazon Aurora MySQL](#)

### Privilèges pour MySQL en tant que base de données cible

Les privilèges requis pour MySQL en tant que cible sont les suivants :

- CRÉER SUR\* . \*
- MODIFIER \* . \*
- DÉPOSEZ \* . \*
- INDEX SUR \* . \*
- RÉFÉRENCES SUR\* . \*
- SELECT ON \*.\*
- CRÉER UNE VUE SUR \* . \*
- SHOW VIEW ON \*.\*
- DÉCLENCHEUR ACTIVÉ\* . \*
- CRÉER UNE ROUTINE SUR\* . \*
- MODIFIER LA ROUTINE SUR \* . \*

- EXÉCUTER SUR\* . \*
- CRÉER DES TABLES TEMPORAIRES SUR\* . \*
- AWS\_LAMBDA\_ACCÈS
- INSÉRER, METTRE À JOUR SUR AWS\_ORACLE\_EXT.\*
- INSÉREZ, METTEZ À JOUR, SUPPRIMEZ SUR AWS\_ORACLE\_EXT\_DATA.\*

Si vous utilisez une base de données MySQL version 5.7 ou inférieure comme cible, accordez l'autorisation INVOKE LAMBDA \*.\* au lieu de AWS\_LAMBDA\_ACCESS. Pour les bases de données MySQL version 8.0 et supérieures, accordez l' AWS\_LAMBDA\_ACCESS autorisation.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON *.* TO 'user_name';
GRANT AWS_LAMBDA_ACCESS TO 'user_name';
GRANT INSERT, UPDATE ON AWS_ORACLE_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_ORACLE_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Si vous utilisez une base de données MySQL version 5.7 ou inférieure comme cible, utilisez GRANT INVOKE LAMBDA ON \*.\* TO '*user\_name*' plutôt que GRANT AWS\_LAMBDA\_ACCESS TO '*user\_name*'.

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants

des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques `DETERMINISTIC`, `READS SQL DATA` ni `NO SQL` pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

## Paramètres de conversion Oracle vers MySQL

Pour modifier les paramètres de conversion d'Oracle vers MySQL, choisissez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Oracle, puis Oracle — MySQL. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Oracle vers MySQL.

Les paramètres de conversion d'Oracle vers MySQL AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour résoudre ce problème, votre base de données Oracle source peut utiliser la ROWID pseudocolonne, mais MySQL ne prend pas en charge des fonctionnalités similaires. AWS SCT peut émuler la ROWID pseudocolonne dans le code converti. Pour ce faire, choisissez Generate as identity pour Generate row ID ? .

Si votre code source Oracle n'utilise pas la ROWID pseudocolonne, choisissez Ne pas générer pour Generate row ID ? Dans ce cas, le code converti fonctionne plus rapidement.

- Pour travailler avec votre code source Oracle lorsqu'il inclut les TO\_NUMBER fonctions TO\_CHAR TO\_DATE, et avec des paramètres que MySQL ne prend pas en charge. Par défaut, AWS SCT émule l'utilisation de ces paramètres dans le code converti.

Lorsque votre code source Oracle inclut uniquement des paramètres pris en charge par PostgreSQL, vous pouvez utiliser TO\_CHAR MySQL TO\_DATE natif et ses fonctions. TO\_NUMBER

Dans ce cas, le code converti fonctionne plus rapidement. Pour inclure uniquement ces paramètres, sélectionnez les valeurs suivantes :

- La fonction TO\_CHAR () n'utilise pas de chaînes de formatage spécifiques à Oracle
- La fonction TO\_DATE () n'utilise pas de chaînes de formatage spécifiques à Oracle
- La fonction TO\_NUMBER () n'utilise pas de chaînes de formatage spécifiques à Oracle
- Pour déterminer si votre base de données et vos applications s'exécutent dans des fuseaux horaires différents. Par défaut, AWS SCT émule les fuseaux horaires dans le code converti. Toutefois, vous n'avez pas besoin de cette émulation lorsque votre base de données et vos applications utilisent le même fuseau horaire. Dans ce cas, sélectionnez Le fuseau horaire du côté client correspond au fuseau horaire du serveur.

## Considérations concernant la migration

Lorsque vous convertissez Oracle en RDS for MySQL ou Aurora MySQL, vous pouvez utiliser une GOTO instruction et une étiquette pour modifier l'ordre dans lequel les instructions s'exécutent. Toutes les instructions PL/SQL qui suivent une GOTO instruction sont ignorées et le traitement se poursuit au niveau de l'étiquette. Vous pouvez utiliser GOTO des instructions et des étiquettes n'importe où dans une procédure, un lot ou un bloc d'instructions. Vous pouvez également suivre les instructions GOTO.

MySQL n'utilise pas d'GOTOinstructions. Lors de la AWS SCT conversion du code contenant une GOTO instruction, il convertit l'instruction pour utiliser une LOOP...END LOOP instruction BEGIN...END or.

Vous trouverez des exemples de AWS SCT conversion d'GOTOinstructions dans le tableau suivant.

Instruction Oracle	Instruction MySQL
<pre> BEGIN   ....   statement1;   ....   GOTO label1;   statement2;   ....   label1:   Statement3;   .... END </pre>	<pre> BEGIN   label1:   BEGIN     ....     statement1;     ....     LEAVE label1;     statement2;     ....   END;   Statement3; </pre>

Instruction Oracle	Instruction MySQL
	<pre>.... END</pre>
<pre>BEGIN .... statement1; .... label1: statement2; .... GOTO label1; statement3; .... statement4; .... END</pre>	<pre>BEGIN .... statement1; .... label1: LOOP statement2; .... ITERATE label1; LEAVE label1; END LOOP; statement3; .... statement4; .... END</pre>
<pre>BEGIN .... statement1; .... label1: statement2; .... statement3; .... statement4; .... END</pre>	<pre>BEGIN .... statement1; .... label1: BEGIN statement2; .... statement3; .... statement4; .... END; END</pre>

## Conversion de l'instruction WITH dans Oracle en RDS pour MySQL ou Amazon Aurora MySQL

Vous utilisez la clause WITH (subquery\_factoring) dans Oracle pour attribuer un nom (query\_name) à un bloc contenant une sous-requête. Vous pouvez ensuite référencer plusieurs endroits de bloc contenant la sous-requête dans la requête en spécifiant query\_name. Si un bloc de sous-requête ne contient ni liens ni paramètres (local, procédure, fonction, package), AWS SCT convertit la clause en vue ou en table temporaire.

L'avantage de convertir la clause en table temporaire est que les références répétées à la sous-requête peuvent être plus efficaces. Cette amélioration de l'efficacité s'explique par le fait que les données sont facilement extraites de la table temporaire au lieu d'être requises par chaque référence. Vous pouvez l'émuler en utilisant des vues supplémentaires ou une table temporaire. Le nom de la vue utilise le format <procedure\_name>\$<subselect\_alias>.

Vous trouverez des exemples dans le tableau ci-dessous.

Instruction Oracle	Instruction MySQL
<pre>CREATE PROCEDURE   TEST_ORA_PG.P_WITH_SELECT_V   ARIABLE_01     (p_state IN NUMBER) AS   l_dept_id NUMBER := 1; BEGIN FOR cur IN       (WITH dept_empl(id, name,         surname,           lastname, state, dept_id)         AS           (             SELECT id, name,               surname,                 lastname, state,                   dept_id             FROM test_ora_               pg.dept_employees             WHERE state =               p_state AND</pre>	<pre>CREATE PROCEDURE test_ora_pg.P_WITH   _SELECT_VARIABLE_01(IN par_P_STATE     DOUBLE) BEGIN   DECLARE var_l_dept_id DOUBLE     DEFAULT 1;   DECLARE var\$id VARCHAR (8000);   DECLARE var\$state VARCHAR (8000);   DECLARE done INT DEFAULT FALSE;   DECLARE cur CURSOR FOR SELECT     ID, STATE   FROM (SELECT     ID, NAME, SURNAME,     LASTNAME, STATE, DEPT_ID     FROM TEST_ORA_PG.DEPT_E     MPLOYEES     WHERE STATE = par_p_sta     te AND DEPT_ID = var_l_dept_id) AS     dept_empl     ORDER BY ID;   DECLARE CONTINUE HANDLER FOR NOT     FOUND</pre>

## Instruction Oracle

```
                dept_id =  
l_dept_id)  
    SELECT id,state  
    FROM dept_emp1  
    ORDER BY id) LOOP  
  
    NULL;  
END LOOP;
```

## Instruction MySQL

```
        SET done := TRUE;  
        OPEN cur;  
  
        read_label:  
        LOOP  
            FETCH cur INTO var$id, var  
$state;  
  
            IF done THEN  
                LEAVE read_label;  
            END IF;  
  
            BEGIN  
            END;  
        END LOOP;  
        CLOSE cur;  
    END;
```

## Instruction Oracle

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_R
  EGULAR_MULT_01
AS
BEGIN

  FOR cur IN (
    WITH dept_emp1 AS
      (
        SELECT id,
name, surname,
          lastname,
state, dept_id
          FROM
test_ora_pg.dept_employees
          WHERE state =
1),
      dept AS
      (SELECT id deptid,
parent_id,
          name deptname
          FROM test_ora_
pg.department
      )
    SELECT dept_emp1
.*,dept.*
          FROM dept_emp1, dept
          WHERE dept_emp1
.dept_id = dept.deptid
      ) LOOP
    NULL;
  END LOOP;

```

## Instruction MySQL

```

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_emp1
` (id, name, surname, lastname, state,
dept_id)
AS
(SELECT id, name, surname, lastname,
state, dept_id
  FROM test_ora_pg.dept_employees
  WHERE state = 1);

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
` (deptid, parent_id,deptname)
AS
(SELECT id deptid, parent_id, name
deptname
  FROM test_ora_pg.department);

CREATE PROCEDURE test_ora_pg.P_WITH
_SELECT_REGULAR_MULT_01()
BEGIN
  DECLARE var$ID DOUBLE;
  DECLARE var$NAME VARCHAR (30);
  DECLARE var$SURNAME VARCHAR (30);
  DECLARE var$LASTNAME VARCHAR (30);
  DECLARE var$STATE DOUBLE;
  DECLARE var$DEPT_ID DOUBLE;
  DECLARE var$deptid DOUBLE;
  DECLARE var$PARENT_ID DOUBLE;
  DECLARE var$deptname VARCHAR
(200);
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT
    dept_emp1.*, dept.*
    FROM TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_emp1
    ` AS dept_emp1,
    TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
    ` AS dept

```



Instruction Oracle	Instruction MySQL
	<pre>WHERE dept_empl.DEPT_ID = dept.DEPTID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur;  read_label: LOOP FETCH cur INTO var\$ID, var\$NAME, var\$SURNAME, var\$LASTNAME, var\$STATE, var \$DEPT_ID, var\$deptid, var\$PARENT_ID, var\$deptname;  IF done THEN LEAVE read_label; END IF;  BEGIN END; END LOOP; CLOSE cur; END;  call test_ora_pg.P_WITH_SELECT_R EGULAR_MULT_01()</pre>

## Instruction Oracle

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_V
  AR_CROSS_02(p_state IN NUMBER)
AS
  l_dept_id NUMBER := 10;
BEGIN
  FOR cur IN (
    WITH emp AS
      (SELECT id, name,
        surname,
          lastname, state,
            dept_id
          FROM test_ora_
pg.dept_employees
        WHERE dept_id >
  10
      ),
      active_emp AS
      (
        SELECT id
          FROM emp
        WHERE emp.state
= p_state
      )
    SELECT *
      FROM active_emp
    ) LOOP
    NULL;
  END LOOP;
END;

```

## Instruction MySQL

```

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_VAR_CROSS_01$emp
  `(id, name, surname, lastname,
    state, dept_id)
AS
(SELECT
  id, name, surname, lastname,
    state, dept_id
  FROM TEST_ORA_PG.DEPT_EMPLOYEES
  WHERE DEPT_ID > 10);

CREATE PROCEDURE
  test_ora_pg.P_WITH_SELECT_V
  AR_CROSS_02(IN par_P_STATE DOUBLE)
BEGIN
  DECLARE var_l_dept_id DOUBLE
  DEFAULT 10;
  DECLARE var$ID DOUBLE;
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT *
    FROM
  (SELECT
    ID
  FROM
    TEST_ORA_
PG.
    `P_WITH_S
    ELECT_VAR_CROSS_01$emp` AS emp
  WHERE emp.STATE = par_p_state)
  AS
  active_emp;
  DECLARE CONTINUE HANDLER FOR NOT
  FOUND
    SET done := TRUE;
  OPEN cur;

  read_label:

```

Instruction Oracle	Instruction MySQL
	<pre>LOOP   FETCH cur INTO var\$ID;    IF done THEN     LEAVE read_label;   END IF;    BEGIN   END; END LOOP; CLOSE cur; END;</pre>

## Migration d'une base de données Oracle vers Amazon RDS for Oracle avec AWS Schema Conversion Tool

Éléments à prendre en compte lors de la migration d'un schéma et de code Oracle vers Amazon RDS pour Oracle :

- AWS SCT peut ajouter des objets de répertoire à l'arborescence des objets. Les objets de répertoire sont des structures logiques qui représentent chacune un répertoire physique sur le système de fichiers du serveur. Vous pouvez utiliser des objets de répertoire avec les packages tels que DBMS\_LOB, UTL\_FILE, DBMS\_FILE\_TRANSFER, l'utilitaire DATAPUMP, et ainsi de suite.
- AWS SCT prend en charge la conversion des tablespaces Oracle en une instance de base de données Amazon RDS for Oracle. Oracle stocke les données logiquement dans les espaces de table et physiquement dans les fichiers de données associés à l'espace de table correspondant. Dans Oracle, vous pouvez créer des espaces de table avec des noms de fichier de données. Amazon RDS prend en charge Oracle Managed Files (OMF) pour les fichiers de données, les fichiers journaux et les fichiers de contrôle uniquement. AWS SCT crée les fichiers de données nécessaires lors de la conversion.
- AWS SCT peut convertir les rôles et les privilèges au niveau du serveur. Le moteur de base de données Oracle utilise une sécurité basée sur le rôle. Un rôle est un ensemble de privilèges que vous pouvez accorder ou révoquer à un utilisateur. Un rôle prédéfini dans Amazon RDS, appelé DBA, autorise normalement tous les privilèges d'administration sur un moteur de base de données

Oracle. Les privilèges suivants ne sont pas disponibles pour le rôle DBA sur une instance de base de données Amazon RDS en utilisant le moteur Oracle :

- Alter database
- Alter system
- Create any directory
- Grant any privilege
- Grant any role
- Create external job

Vous pouvez accorder tous les autres privilèges à un rôle d'utilisateur Amazon RDS pour Oracle, y compris les privilèges de filtrage avancés et de colonne.

- AWS SCT prend en charge la conversion de tâches Oracle en tâches pouvant être exécutées sur Amazon RDS for Oracle. Quelques limitations s'appliquent à la conversion, notamment :
  - Les tâches exécutables ne sont pas prises en charge.
  - Les tâches de planification qui utilisent le type de données ANYDATA comme argument ne sont pas prises en charge.
- Oracle Real Application Clusters (RAC) One Node est une option pour Oracle Database Enterprise Edition qui a été introduite avec Oracle Database 11g version 2. Amazon RDS pour Oracle ne prend pas en charge la fonction RAC. Pour un haut niveau de disponibilité, utilisez multi-AZ Amazon RDS.

Dans un déploiement multi-AZ, Amazon RDS fournit et maintient automatiquement un réplica de secours synchrone dans une autre zone de disponibilité. L'instance de base de données principale est répliquée de manière synchrone entre les zones de disponibilité et un réplica de secours. Cette fonctionnalité offre la redondance des données, élimine les figements d'E/S et minimise les pics de latence pendant les sauvegardes du système.

- Oracle Spatial fournit un schéma SQL et des fonctions qui facilitent le stockage, la récupération, la mise à jour et la requête de collections de données spatiales dans une base de données Oracle. Oracle Locator fournit des capacités généralement requises pour prendre en charge les applications basées sur des services sans fil et sur Internet, ainsi que les solutions GIS basées sur le partenariat. Oracle Locator est un sous-ensemble limité d'Oracle Spatial.

Pour utiliser les fonctionnalités Oracle Spatial et Oracle Locator, ajoutez l'option SPATIAL ou LOCATOR (mutuellement exclusives) au groupe d'options de votre instance de base de données.

Il existe certains prérequis pour utiliser Oracle Spatial et Oracle Locator sur une instance de base de données Amazon RDS pour Oracle :

- L'instance doit utiliser Oracle Enterprise Edition version 12.1.0.2.v6 ou supérieure, ou 11.2.0.4.v10 ou supérieure.
- Votre instance doit être dans un VPC (cloud privé virtuel).
- L'instance doit être la classe d'instance de base de données qui peut prendre en charge la fonctionnalité Oracle. Par exemple, Oracle Spatial n'est pas pris en charge pour les classes d'instance de base de données db.m1.small, db.t1.micro, db.t2.micro ou db.t2.small. Pour plus d'informations, consultez la section [Support des classes d'instance de base de données pour Oracle](#).
- L'option Mise à niveau automatique de versions mineures doit être activée pour l'instance. Amazon RDS met à jour votre instance de base de données vers le PSU Oracle le plus récent en cas de vulnérabilités de sécurité présentant un score CVSS égal ou supérieur à 9, ou d'autres vulnérabilités de sécurité annoncées. Pour plus d'informations, veuillez consulter la rubrique

#### [Paramètres pour les instances de base de données Oracle](#).

- Si votre instance de base de données est de version 11.2.0.4.v10 ou supérieure, vous devez installer l'option XMLDB. Pour plus d'informations, veuillez consulter la rubrique

#### Base de [données XML Oracle](#).

- Vous devez disposer d'une licence Oracle Spatial fournie par Oracle. Pour plus d'informations, consultez [Oracle Spatial et Graph](#) dans la documentation d'Oracle.
- Data Guard est inclus avec Oracle Database Enterprise Edition. Pour un haut niveau de disponibilité, utilisez la fonction multi-AZ Amazon RDS.

Dans un déploiement multi-AZ, Amazon RDS fournit et maintient automatiquement un réplica de secours synchrone dans une autre zone de disponibilité. L'instance de base de données principale est répliquée de manière synchrone entre les zones de disponibilité et un réplica de secours. Cette fonctionnalité offre la redondance des données, élimine les figements d'E/S et minimise les pics de latence pendant les sauvegardes du système.

- AWS SCT prend en charge la conversion des objets Oracle DBMS\_SCHEDULER lors de la migration vers Amazon RDS for Oracle. Le rapport AWS SCT d'évaluation indique si un objet de planification peut être converti. Pour plus d'informations sur l'utilisation des objets de planification avec Amazon RDS, consultez la [documentation Amazon RDS](#).

- Pour les conversions d'Oracle vers Amazon RDS pour Oracle, les liens de base de données sont pris en charge. Un lien de base de données est un objet de schéma dans une base de données qui vous permet d'accéder aux objets d'une autre base de données. L'autre base de données ne doit pas nécessairement être une base de données Oracle. Toutefois, pour accéder aux bases de données autres qu'Oracle, vous devez utiliser Oracle Heterogeneous Services.

Une fois que vous créez un lien de base de données, vous pouvez utiliser le lien dans les instructions SQL pour faire référence à des tables, des vues et des objets PL/SQL dans l'autre base de données. Pour utiliser un lien de base de données, ajoutez @dbLink à la table, à la vue ou au nom de l'objet PL/SQL. Vous pouvez interroger une table ou une vue dans l'autre base de données avec l'instruction SELECT. Pour plus d'informations sur l'utilisation des liens de base de données Oracle, consultez la [documentation Oracle](#).

Pour plus d'informations sur l'utilisation des liens de base de données avec Amazon RDS, consultez la [documentation Amazon RDS](#).

- Le rapport AWS SCT d'évaluation fournit les indicateurs du serveur pour la conversion. Ces metrics sur votre instance Oracle sont les suivantes :
  - Capacité de calcul et de mémoire de l'instance de base de données cible.
  - Fonctionnalités Oracle non prises en charge, telles que Real Application Clusters, qu'Amazon RDS ne prend pas en charge.
  - Charge de travail en lecture/écriture sur disque
  - Total moyen du débit de disque
  - Les informations du serveur telles que le nom du serveur, le nom d'hôte, le système d'exploitation et le jeu de caractères.

## Privilèges accordés à RDS for Oracle en tant que cible

Pour migrer vers Amazon RDS for Oracle, créez un utilisateur de base de données privilégié. Vous pouvez utiliser l'exemple de code suivant.

```
CREATE USER user_name IDENTIFIED BY your_password;  
  
-- System privileges  
GRANT DROP ANY CUBE BUILD PROCESS TO user_name;  
GRANT ALTER ANY CUBE TO user_name;  
GRANT CREATE ANY CUBE DIMENSION TO user_name;  
GRANT CREATE ANY ASSEMBLY TO user_name;
```

```
GRANT ALTER ANY RULE TO user_name;  
GRANT SELECT ANY DICTIONARY TO user_name;  
GRANT ALTER ANY DIMENSION TO user_name;  
GRANT CREATE ANY DIMENSION TO user_name;  
GRANT ALTER ANY TYPE TO user_name;  
GRANT DROP ANY TRIGGER TO user_name;  
GRANT CREATE ANY VIEW TO user_name;  
GRANT ALTER ANY CUBE BUILD PROCESS TO user_name;  
GRANT CREATE ANY CREDENTIAL TO user_name;  
GRANT DROP ANY CUBE DIMENSION TO user_name;  
GRANT DROP ANY ASSEMBLY TO user_name;  
GRANT DROP ANY PROCEDURE TO user_name;  
GRANT ALTER ANY PROCEDURE TO user_name;  
GRANT ALTER ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT DROP ANY MEASURE FOLDER TO user_name;  
GRANT CREATE ANY MEASURE FOLDER TO user_name;  
GRANT DROP ANY CUBE TO user_name;  
GRANT DROP ANY MINING MODEL TO user_name;  
GRANT CREATE ANY MINING MODEL TO user_name;  
GRANT DROP ANY EDITION TO user_name;  
GRANT CREATE ANY EVALUATION CONTEXT TO user_name;  
GRANT DROP ANY DIMENSION TO user_name;  
GRANT ALTER ANY INDEXTYPE TO user_name;  
GRANT DROP ANY TYPE TO user_name;  
GRANT CREATE ANY PROCEDURE TO user_name;  
GRANT CREATE ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT CREATE ANY CUBE TO user_name;  
GRANT COMMENT ANY MINING MODEL TO user_name;  
GRANT ALTER ANY MINING MODEL TO user_name;  
GRANT DROP ANY SQL PROFILE TO user_name;  
GRANT CREATE ANY JOB TO user_name;  
GRANT DROP ANY EVALUATION CONTEXT TO user_name;  
GRANT ALTER ANY EVALUATION CONTEXT TO user_name;  
GRANT CREATE ANY INDEXTYPE TO user_name;  
GRANT CREATE ANY OPERATOR TO user_name;  
GRANT CREATE ANY TRIGGER TO user_name;  
GRANT DROP ANY ROLE TO user_name;  
GRANT DROP ANY SEQUENCE TO user_name;  
GRANT DROP ANY CLUSTER TO user_name;  
GRANT DROP ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT ALTER ANY ASSEMBLY TO user_name;  
GRANT CREATE ANY RULE SET TO user_name;  
GRANT ALTER ANY OUTLINE TO user_name;  
GRANT UNDER ANY TYPE TO user_name;
```

```
GRANT CREATE ANY TYPE TO user_name;  
GRANT DROP ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY ROLE TO user_name;  
GRANT DROP ANY VIEW TO user_name;  
GRANT ALTER ANY INDEX TO user_name;  
GRANT COMMENT ANY TABLE TO user_name;  
GRANT CREATE ANY TABLE TO user_name;  
GRANT CREATE USER TO user_name;  
GRANT DROP ANY RULE SET TO user_name;  
GRANT CREATE ANY CONTEXT TO user_name;  
GRANT DROP ANY INDEXTYPE TO user_name;  
GRANT ALTER ANY OPERATOR TO user_name;  
GRANT CREATE ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY SEQUENCE TO user_name;  
GRANT DROP ANY SYNONYM TO user_name;  
GRANT CREATE ANY SYNONYM TO user_name;  
GRANT DROP USER TO user_name;  
GRANT ALTER ANY MEASURE FOLDER TO user_name;  
GRANT ALTER ANY EDITION TO user_name;  
GRANT DROP ANY RULE TO user_name;  
GRANT CREATE ANY RULE TO user_name;  
GRANT ALTER ANY RULE SET TO user_name;  
GRANT CREATE ANY OUTLINE TO user_name;  
GRANT UNDER ANY TABLE TO user_name;  
GRANT UNDER ANY VIEW TO user_name;  
GRANT DROP ANY DIRECTORY TO user_name;  
GRANT ALTER ANY CLUSTER TO user_name;  
GRANT CREATE ANY CLUSTER TO user_name;  
GRANT ALTER ANY TABLE TO user_name;  
GRANT CREATE ANY CUBE BUILD PROCESS TO user_name;  
GRANT ALTER ANY CUBE DIMENSION TO user_name;  
GRANT CREATE ANY EDITION TO user_name;  
GRANT CREATE ANY SQL PROFILE TO user_name;  
GRANT ALTER ANY SQL PROFILE TO user_name;  
GRANT DROP ANY OUTLINE TO user_name;  
GRANT DROP ANY CONTEXT TO user_name;  
GRANT DROP ANY OPERATOR TO user_name;  
GRANT DROP ANY LIBRARY TO user_name;  
GRANT ALTER ANY LIBRARY TO user_name;  
GRANT CREATE ANY LIBRARY TO user_name;  
GRANT ALTER ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY TRIGGER TO user_name;  
GRANT CREATE ANY SEQUENCE TO user_name;  
GRANT DROP ANY INDEX TO user_name;
```



```
GRANT CREATE ANY INDEX TO user_name;  
GRANT DROP ANY TABLE TO user_name;  
GRANT SELECT_CATALOG_ROLE TO user_name;  
GRANT SELECT ANY SEQUENCE TO user_name;  
  
-- Database Links  
GRANT CREATE DATABASE LINK TO user_name;  
GRANT CREATE PUBLIC DATABASE LINK TO user_name;  
GRANT DROP PUBLIC DATABASE LINK TO user_name;  
  
-- Server Level Objects (directory)  
GRANT CREATE ANY DIRECTORY TO user_name;  
GRANT DROP ANY DIRECTORY TO user_name;  
-- (for RDS only)  
GRANT EXECUTE ON RDSADMIN.RDSADMIN_UTIL TO user_name;  
  
-- Server Level Objects (tablespace)  
GRANT CREATE TABLESPACE TO user_name;  
GRANT DROP TABLESPACE TO user_name;  
  
-- Server Level Objects (user roles)  
/* (grant source privileges with admin option or convert roles/privs as DBA) */  
  
-- Queues  
grant execute on DBMS_AQADM to user_name;  
grant aq_administrator_role to user_name;  
  
-- for Materialized View Logs creation  
GRANT SELECT ANY TABLE TO user_name;  
  
-- Roles  
GRANT RESOURCE TO user_name;  
GRANT CONNECT TO user_name;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

## Limitations lors de la conversion d'Oracle vers Amazon RDS for Oracle

Quelques éléments à prendre en compte lors de la migration d'un schéma Oracle et d'un code vers Amazon RDS pour Oracle :

- Un rôle prédéfini dans Amazon RDS, appelé DBA, autorise normalement tous les privilèges d'administration sur un moteur de base de données Oracle. Les privilèges suivants ne sont pas disponibles pour le rôle DBA sur une instance de base de données Amazon RDS en utilisant le moteur Oracle :
  - Alter database
  - Alter system
  - Create any directory
  - Grant any privilege
  - Grant any role
  - Create external job

Vous pouvez accorder tous les autres privilèges à un rôle d'utilisateur Oracle RDS.

- Amazon RDS for Oracle prend en charge l'audit traditionnel, l'audit détaillé à l'aide du package DBMS\_FGA et Oracle Unified Auditing.
- Amazon RDS pour Oracle ne prend pas en charge la capture des données modifiées (CDC). Pour effectuer le CDC pendant et après une migration de base de données, utilisez AWS Database Migration Service.

## Connexion aux bases de données PostgreSQL à l'aide du AWS Schema Conversion Tool

Vous pouvez les utiliser AWS SCT pour convertir des schémas, des objets de code de base de données et du code d'application de PostgreSQL vers les cibles suivantes :

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Pour plus d'informations, consultez les sections suivantes :

### Rubriques

- [Privilèges pour PostgreSQL en tant que base de données source](#)
- [Connexion à PostgreSQL en tant que source](#)

- [Privilèges pour MySQL en tant que base de données cible](#)

## Privilèges pour PostgreSQL en tant que base de données source

Les privilèges requis pour PostgreSQL en tant que source sont les suivants :

- CONNEXION À LA BASE DE DONNÉES *<database\_name>*
- UTILISATION SUR LE SCHÉMA *<database\_name>*
- SÉLECTIONNEZ SUR TOUTES LES TABLES DU SCHÉMA *<database\_name>*
- SÉLECTIONNEZ TOUTES LES SÉQUENCES DU SCHÉMA *<database\_name>*

## Connexion à PostgreSQL en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source PostgreSQL avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source PostgreSQL

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez PostgreSQL, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de la connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source PostgreSQL, suivez les instructions suivantes :

Paramètre	Action
Server name	<p>Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.</p> <p>Vous pouvez vous connecter à votre base de données PostgreSQL source à l'aide IPv6 d'un protocole d'adresse . Pour ce faire, assurez-vous d'utiliser des crochets pour saisir l'adresse IP, comme indiqué dans l'exemple suivant.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Tapez le nom de la base de données PostgreSQL.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans un projet. Pour éviter d'exposer le mot de passe de votre base de données source, AWS SCT ne le stocke pas par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Vérifier le certificat du serveur : sélectionnez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li><li>• Trust store : emplacement d'un trust store contenant des certificats. Pour que cet emplacement apparaisse dans la section Paramètres généraux, assurez-vous de l'ajouter.</li></ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. L'activation de cette option vous permet de stocker le mot de passe de la base de données et de vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
chemin du pilote PostgreSQL	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Privilèges pour MySQL en tant que base de données cible

Les privilèges requis pour MySQL en tant que cible lors de la migration depuis PostgreSQL sont les suivants :

- CRÉER SUR\* . \*
- MODIFIER \* . \*
- DÉPOSEZ \* . \*
- INDEX SUR \* . \*
- RÉFÉRENCES SUR\* . \*
- SELECT ON \*.\*
- CRÉER UNE VUE SUR \* . \*
- SHOW VIEW ON \*.\*
- DÉCLENCHER ACTIVÉ\* . \*
- CRÉER UNE ROUTINE SUR\* . \*
- MODIFIER LA ROUTINE SUR \* . \*
- EXÉCUTER SUR\* . \*
- INSÉRER, METTRE À JOUR SUR AWS\_POSTGRESQL\_EXT. \*
- INSÉREZ, METTEZ À JOUR, SUPPRIMEZ SUR AWS\_POSTGRESQL\_EXT\_DATA. \*
- CRÉEZ DES TABLES TEMPORAIRES SUR AWS\_POSTGRESQL\_EXT\_DATA. \*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
```

```
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_POSTGRESQL_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_POSTGRESQL_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_POSTGRESQL_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques `DETERMINISTIC`, `READS SQL DATA` ni `NO SQL` pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

## Connexion aux bases de données SAP à l'aide du AWS Schema Conversion Tool

Vous pouvez les utiliser AWS SCT pour convertir les schémas, les objets de code de base de données et le code d'application de SAP (Sybase) Adaptive Server Enterprise (ASE) vers les cibles suivantes :

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for MariaDB
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Pour plus d'informations, consultez les sections suivantes :

### Rubriques

- [Privilèges pour SAP ASE en tant que base de données source](#)
- [Connexion à SAP ASE \(Sybase\) en tant que source](#)
- [Privilèges pour MySQL en tant que base de données cible](#)

- [Paramètres de conversion SAP ASE vers MySQL](#)
- [Privilèges pour PostgreSQL en tant que base de données cible](#)
- [Paramètres de conversion de SAP ASE vers PostgreSQL](#)

## Privilèges pour SAP ASE en tant que base de données source

Pour utiliser une base de données SAP ASE comme source, vous devez créer un utilisateur de base de données et accorder des autorisations. Pour ce faire, suivez les étapes ci-dessous.

### Création et configuration d'un utilisateur de base de données

1. Etablit une connexion à la base de données source.
2. Créez un utilisateur de base de données à l'aide des commandes suivantes. Entrez un mot de passe pour le nouvel utilisateur.

```
USE master
CREATE LOGIN min_privs WITH PASSWORD <password>
sp_adduser min_privs
grant select on dbo.spt_values to min_privs
grant select on asehostname to min_privs
```

3. Pour chaque base de données que vous allez migrer, accordez les privilèges suivants.

```
USE <database_name>
sp_adduser min_privs
grant select on dbo.sysusers to min_privs
grant select on dbo.sysobjects to min_privs
grant select on dbo.sysindexes to min_privs
grant select on dbo.syscolumns to min_privs
grant select on dbo.sysreferences to min_privs
grant select on dbo.syscomments to min_privs
grant select on dbo.syspartitions to min_privs
grant select on dbo.syspartitionkeys to min_privs
grant select on dbo.sysconstraints to min_privs
grant select on dbo.systypes to min_privs
grant select on dbo.sysqueryplans to min_privs
```



## Connexion à SAP ASE (Sybase) en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source SAP ASE avec le AWS Schema Conversion Tool.

Pour vous connecter à une base de données source SAP ASE

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez SAP ASE, puis Next.


La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source SAP ASE, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.

Paramètre	Action
Database (Base de données)	Entrez le nom de la base de données SAP ASE.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <div data-bbox="656 527 1507 1129" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez votre AWS SCT projet et que vous le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p></div>
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Vérifier le certificat du serveur : sélectionnez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li><li>• Trust store : emplacement d'un trust store contenant des certificats.</li></ul>

Paramètre	Action
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. L'activation de cette option vous permet de stocker le mot de passe de la base de données et de vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.
Chemin du pilote SAP ASE	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Privilèges pour MySQL en tant que base de données cible

Les privilèges requis pour MySQL en tant que cible sont les suivants :

- CRÉER SUR\* . \*
- MODIFIER\* . \*
- DÉPOSEZ\* . \*
- INDEX SUR\* . \*
- RÉFÉRENCES SUR\* . \*
- SELECT ON\*.\*
- CRÉER UNE VUE SUR\* . \*
- SHOW VIEW ON\*.\*

- DÉCLENCHER ACTIVÉ\* . \*
- CRÉER UNE ROUTINE SUR\* . \*
- MODIFIER LA ROUTINE SUR \* . \*
- EXÉCUTER SUR\* . \*
- INSÉRER, METTRE À JOUR SUR AWS\_SAPASE\_EXT. \*
- CRÉER DES TABLES TEMPORAIRES SUR AWS\_SAPASE\_EXT. \*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_SAPASE_EXT.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_SAPASE_EXT.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques DETERMINISTIC, READS SQL DATA ni NO SQL pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.

## Paramètres de conversion SAP ASE vers MySQL

Pour modifier les paramètres de conversion SAP ASE vers MySQL, choisissez Paramètres, puis Paramètres de conversion. Dans la liste supérieure, choisissez SAP ASE, puis SAP ASE — MySQL ou SAP ASE — Amazon Aurora (compatible avec MySQL). AWS SCT affiche tous les paramètres disponibles pour la conversion de SAP ASE vers PostgreSQL.

Les paramètres de conversion de SAP ASE vers MySQL AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la sévérité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour utiliser les noms exacts des objets de base de données source dans le code converti.

Par défaut, AWS SCT convertit les noms des objets, des variables et des paramètres de base de données en minuscules. Pour conserver les majuscules et minuscules d'origine de ces noms, sélectionnez Traiter les noms des objets de base de données source en distinguant les majuscules et minuscules. Choisissez cette option si vous utilisez des noms d'objets distinguant majuscules et minuscules dans votre serveur de base de données SAP ASE source.

## Privilèges pour PostgreSQL en tant que base de données cible

Pour utiliser PostgreSQL comme cible AWS SCT , le privilège est requis. CREATE ON DATABASE Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par "\$user", public\_synonyms, public.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';
```

```
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Dans PostgreSQL, seul le propriétaire du schéma ou un `superuser` peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

## Paramètres de conversion de SAP ASE vers PostgreSQL

Pour modifier les paramètres de conversion de SAP ASE vers PostgreSQL, choisissez Paramètres, puis Paramètres de conversion. Dans la liste supérieure, choisissez SAP ASE, puis SAP ASE — PostgreSQL ou SAP ASE — Amazon Aurora (compatible avec PostgreSQL). AWS SCT affiche tous les paramètres disponibles pour la conversion de SAP ASE vers PostgreSQL.

Les paramètres de conversion de SAP ASE vers PostgreSQL incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la sévérité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le modèle à utiliser pour les noms de schéma dans le code converti. Pour le modèle de génération de nom de schéma, choisissez l'une des options suivantes :
  - `<source_db>`— Utilisez le nom de base de données SAP ASE comme nom de schéma dans PostgreSQL.

- `<source_schema>`— Utilise le nom du schéma SAP ASE comme nom de schéma dans PostgreSQL.
- `_ <source_db><schema>`— Utilise une combinaison de la base de données SAP ASE et des noms de schéma comme nom de schéma dans PostgreSQL.
- Pour utiliser les noms exacts des objets de base de données source dans le code converti.

Par défaut, AWS SCT convertit les noms des objets, des variables et des paramètres de base de données en minuscules. Pour conserver les majuscules et minuscules d'origine de ces noms, sélectionnez Traiter les noms des objets de base de données source en distinguant les majuscules et minuscules. Choisissez cette option si vous utilisez des noms d'objets distinguant majuscules et minuscules dans votre serveur de base de données SAP ASE source.

Pour les opérations distinguant majuscules et minuscules, AWS SCT vous pouvez éviter la conversion des noms d'objets de base de données en minuscules. Pour ce faire, sélectionnez Éviter de convertir en minuscules pour les opérations sensibles aux majuscules et minuscules.

- Permettre l'utilisation d'index portant le même nom dans différentes tables dans SAP ASE.

Dans PostgreSQL, tous les noms d'index que vous utilisez dans le schéma doivent être uniques. Pour vous assurer que cela AWS SCT génère des noms uniques pour tous vos index, sélectionnez Générer des noms uniques pour les index.

## Connect Microsoft SQL Server avec AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code de base de données et du code d'application depuis SQL Server vers les cibles suivantes :

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for SQL Server
- Amazon RDS for MariaDB

**Note**

AWS SCT ne prend pas en charge l'utilisation d'Amazon RDS pour SQL Server comme source.

Vous pouvez l'utiliser AWS SCT pour créer un rapport d'évaluation pour la migration de schémas, d'objets de code de base de données et de code d'application de SQL Server vers BabelFish pour Aurora PostgreSQL, comme décrit ci-dessous.

### Rubriques

- [Privilèges pour Microsoft SQL Server en tant que source](#)
- [Utilisation de l'authentification Windows lors de l'utilisation de Microsoft SQL Server comme source](#)
- [Connexion à SQL Server en tant que source](#)
- [Conversion de SQL Server en MySQL](#)
- [Migration de SQL Server vers PostgreSQL avec AWS Schema Conversion Tool](#)
- [Migration de SQL Server vers Amazon RDS for SQL Server avec AWS Schema Conversion Tool](#)

## Privilèges pour Microsoft SQL Server en tant que source

Les privilèges requis pour Microsoft SQL Server en tant que source sont les suivants :

- VIEW DEFINITION
- VIEW DATABASE STATE

Ce VIEW DEFINITION privilège permet aux utilisateurs disposant d'un accès public de consulter les définitions des objets. AWS SCT utilise le VIEW DATABASE STATE privilège pour vérifier les fonctionnalités de l'édition SQL Server Enterprise.

Répétez l'octroi pour chaque base de données dont vous convertissez le schéma.

En outre, accordez les privilèges suivants sur la base de données master :

- VIEW SERVER STATE
- VIEW ANY DEFINITION



AWS SCT utilise le VIEW SERVER STATE privilège pour collecter les paramètres et la configuration du serveur. Assurez-vous d'accorder le VIEW ANY DEFINITION privilège d'afficher les points de terminaison.

Pour lire des informations sur Microsoft Analysis Services, exécutez la commande suivante sur la base de données master.

```
EXEC master..sp_addsrvrolemember @loginame = N'<user_name>', @rolename = N'sysadmin'
```

Dans l'exemple précédent, remplacez l'<user\_name> espace réservé par le nom de l'utilisateur auquel vous avez accordé les privilèges précédents.

Pour lire des informations sur l'agent SQL Server, ajoutez votre utilisateur au SQLAgentUser rôle. Exécutez la commande suivante sur la base de données msdb.

```
EXEC sp_addrolemember <SQLAgentRole>, <user_name>;
```

Dans l'exemple précédent, remplacez l'espace réservé <SQLAgentRole> par le nom du rôle SQL Server Agent. Remplacez ensuite l'<user\_name> espace réservé par le nom de l'utilisateur auquel vous avez accordé les privilèges précédents. Pour plus d'informations, consultez la section [Ajouter un utilisateur au rôle SQLAgent d'utilisateur](#) dans le guide de l'utilisateur Amazon RDS.

Pour détecter l'expédition de journaux, accordez le privilège SELECT on dbo.log\_shipping\_primary\_databases sur la base de données msdb.

Pour utiliser l'approche de notification de la réplication DDL, accordez le RECEIVE ON <schema\_name>.<queue\_name> privilège sur vos bases de données sources. Dans cet exemple, remplacez l'espace réservé <schema\_name> par le nom du schéma de la base de données. Remplacez ensuite l'espace réservé <queue\_name> par le nom d'une table de file d'attente.

## Utilisation de l'authentification Windows lors de l'utilisation de Microsoft SQL Server comme source

Si votre application s'exécute sur un intranet Windows, vous pouvez utiliser l'authentification Windows pour accéder à la base de données. L'authentification Windows utilise l'identité Windows actuelle établie sur le thread du système d'exploitation pour accéder à la base de données SQL Server. Vous pouvez ensuite faire correspondre l'identité Windows à une base de données et à des permissions

SQL Server. Pour vous connecter à SQL Server à l'aide de l'authentification Windows, vous devez spécifier l'identité Windows qu'utilise votre application. Vous devez également accorder l'accès à l'identité Windows à la base de données SQL Server.

SQL Server possède deux modes d'accès : le mode d'authentification Windows et le mode mixte. Le mode d'authentification Windows active l'authentification Windows et désactive l'authentification SQL Server. Le mode mixte active à la fois l'authentification Windows et l'authentification SQL Server. L'authentification Windows est toujours disponible et ne peut pas être désactivée. Pour plus d'informations sur l'authentification Windows, consultez la documentation Microsoft Windows.

L'exemple ci-dessous illustre un moyen de créer un utilisateur dans TEST\_DB.

```
USE [TEST_DB]
CREATE USER [TestUser] FOR LOGIN [TestDomain\TestUser]
GRANT VIEW DEFINITION TO [TestUser]
GRANT VIEW DATABASE STATE TO [TestUser]
```

## Utilisation de l'authentification Windows avec une connexion JDBC

Le pilote JDBC ne prend pas en charge l'authentification Windows lorsque le pilote est utilisé sur des systèmes d'exploitation autres que Windows. Les informations d'authentification Windows, telles que le nom d'utilisateur et le mot de passe, ne sont pas spécifiées automatiquement lors de la connexion à SQL Server à partir de systèmes d'exploitation autres que Windows. Dans ce cas, les applications doivent plutôt utiliser l'authentification SQL Server.

Dans la chaîne de connexion JDBC, le paramètre `integratedSecurity` doit être spécifié pour se connecter à l'aide de l'authentification Windows. Le pilote JDBC prend en charge l'authentification Windows intégrée sur les systèmes d'exploitation Windows via le paramètre de chaîne de connexion `integratedSecurity`.

Pour utiliser l'authentification intégrée

1. Installez le pilote JDBC.
2. Copiez le fichier `sqljdbc_auth.dll` dans un répertoire sur le chemin d'accès système Windows sur l'ordinateur où le pilote JDBC est installé.

Les `sqljdbc_auth.dll` fichiers sont installés à l'emplacement suivant :

<répertoire d'installation>\sqljdbc\_<version>\<language>\auth\

Lorsque vous essayez d'établir une connexion à la base de données SQL Server à l'aide de l'authentification Windows, vous pouvez obtenir l'erreur suivante : Ce pilote n'est pas configuré pour l'authentification intégrée. Ce problème peut être résolu à l'aide des actions suivantes :

- Déclaration de deux variables qui pointent vers le chemin d'accès de votre installation JDBC :

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu(où se trouve  
votre fichier sqljdbc4.jar) ;
```

```
variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu\auth  
\x86(si vous utilisez un système d'exploitation 32 bits) ou D\lib\JDBC4.1\enu\auth\x64 (si  
vous utilisez un système d'exploitation 64 bits). C'est ici que se sqljdbc_auth.dll trouve le  
vôtre.
```

- Copiez dans `sqljdbc_auth.dll` le dossier dans lequel votre JDK/JRE est en cours d'exécution. Vous pouvez effectuer la copie vers le dossier `lib`, dans le dossier `Bin`, etc. Par exemple, vous pouvez copier le fichier dans le dossier suivant.

```
[JDK_INSTALLED_PATH]\bin;  
[JDK_INSTALLED_PATH]\jre\bin;  
[JDK_INSTALLED_PATH]\jre\lib;  
[JDK_INSTALLED_PATH]\lib;
```

- Assurez-vous que le dossier de votre bibliothèque JDBC ne contient que le `SQLJDBC4` fichier `.jar`. Supprimez tous les autres fichiers `sqljdbc*.jar` de ce dossier (ou copiez-les dans un autre dossier). Si vous ajoutez le pilote dans le cadre de votre programme, assurez-vous d'ajouter uniquement `SQLJDBC4 .jar` comme pilote à utiliser.
- Copie du fichier `sqljdbc_auth.dll` dans le dossier contenant votre application.

#### Note

Si vous exécutez une machine virtuelle Java (JVM) 32 bits, utilisez le fichier `sqljdbc_auth.dll` situé dans le dossier `x86`, même si la version du système d'exploitation est `x64`. Si vous exécutez une machine virtuelle Java 64 bits sur un processeur `x64`, utilisez le fichier `sqljdbc_auth.dll` dans le dossier `x64`.

Lorsque vous vous connectez à une base de données SQL Server, vous pouvez choisir l'authentification Windows ou l'authentification SQL Server comme option d'authentification.

## Connexion à SQL Server en tant que source

Utilisez la procédure suivante pour vous connecter à une base de données source Microsoft SQL Server avec le AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Microsoft SQL Server

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Microsoft SQL Server, puis Next.


La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Microsoft SQL Server, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du service DNS (Domain Name Service) ou l'adresse IP du serveur de la base de données source.  Vous pouvez vous connecter à votre base de données SQL Server source à l'aide d'un protocole d'IPv6 adresse. Pour ce faire, assurez-vous d'utiliser des crochets pour saisir l'adresse IP, comme indiqué dans l'exemple suivant.

Paramètre	Action
	
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Instance name	Tapez le nom de l'instance de la base de données SQL Server. Pour trouver le nom de l'instance, exécutez la requête <code>SELECT @@servername;</code> sur la base de données SQL Server.
Authentification	Choisissez le type d'authentification dans Authentification Windows et Authentification SQL Server.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez votre AWS SCT projet puis le rouvrez, vous êtes invité à saisir le mot de passe pour vous connecter à votre base de données source selon vos besoins.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Certificat de serveur de confiance : sélectionnez cette option pour approuver le certificat de serveur.</li> <li>• Trust store : emplacement d'un trust store contenant des certificats. Pour que cet emplacement apparaisse dans la section Paramètres généraux, assurez-vous de l'ajouter.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. L'activation de cette option vous permet de stocker le mot de passe de la base de données et de vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Sql Server Driver Path	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>
Bibliothèque d'authentification Windows	<p>Entrez le chemin d'accès au sqljdbc_auth.dll fichier. Par défaut, ce fichier est installé à l'emplacement suivant :</p> <p><i>&lt;installation directory of the JDBC driver&gt;sqljdbc_&lt;version&gt; \&lt;language&gt; \auth\</i></p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.

6. Choisissez Connect pour vous connecter à votre base de données source.

## Conversion de SQL Server en MySQL

Pour émuler les fonctions de base de données Microsoft SQL Server dans votre code MySQL converti, utilisez le pack d'extension SQL Server vers MySQL intégré. AWS SCT Pour plus d'informations sur les packs d'extension, consultez [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

### Rubriques

- [Privilèges pour MySQL en tant que base de données cible](#)
- [Paramètres de conversion SQL Server vers MySQL](#)
- [Considérations concernant la migration](#)

### Privilèges pour MySQL en tant que base de données cible

Les privilèges requis pour MySQL en tant que cible sont les suivants :

- CRÉER SUR\* . \*
- MODIFIER \* . \*
- DÉPOSEZ \* . \*
- INDEX SUR \* . \*
- RÉFÉRENCES SUR\* . \*
- SELECT ON \*.\*
- CRÉER UNE VUE SUR \* . \*
- SHOW VIEW ON \*.\*
- DÉCLENCHEUR ACTIVÉ\* . \*
- CRÉER UNE ROUTINE SUR\* . \*
- MODIFIER LA ROUTINE SUR \* . \*
- EXÉCUTER SUR\* . \*
- INSÉRER, METTRE À JOUR SUR AWS\_SQLSERVER\_EXT. \*
- INSÉREZ, METTEZ À JOUR, SUPPRIMEZ SUR AWS\_SQLSERVER\_EXT\_DATA. \*

- CRÉEZ DES TABLES TEMPORAIRES SUR AWS\_SQLSERVER\_EXT\_DATA. \*

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_SQLSERVER_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

Si vous utilisez une base de données MySQL version 5.7 ou inférieure comme cible, exécutez la commande suivante. Pour les bases de données MySQL version 8.0 et supérieures, cette commande est obsolète.

```
GRANT SELECT ON mysql.proc TO 'user_name';
```

Pour utiliser Amazon RDS for MySQL ou Aurora MySQL en tant que cible, définissez le paramètre `lower_case_table_names` sur 1. Cette valeur signifie que le serveur MySQL traite les identifiants des noms d'objets tels que les tables, les index, les déclencheurs et les bases de données sans distinction entre majuscules et minuscules. Si vous avez activé la journalisation binaire dans votre instance cible, définissez le paramètre `log_bin_trust_function_creators` sur 1. Dans ce cas, vous n'avez pas besoin d'utiliser les caractéristiques DETERMINISTIC, READS SQL DATA ni NO SQL pour créer des fonctions stockées. Pour configurer ces paramètres, créez un nouveau groupe de paramètres de base de données ou modifiez un groupe de paramètres de base de données existant.



## Paramètres de conversion SQL Server vers MySQL

Pour modifier les paramètres de conversion de SQL Server vers MySQL, AWS SCT sélectionnez Paramètres, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez SQL Server, puis SQL Server — MySQL. AWS SCT affiche tous les paramètres disponibles pour la conversion de SQL Server vers MySQL.

Les paramètres de conversion de SQL Server vers MySQL AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour permettre à votre base de données SQL Server source de stocker le résultat de EXEC dans une table. AWS SCT crée des tables temporaires et une procédure supplémentaire pour émuler cette fonctionnalité. Pour utiliser cette émulation, sélectionnez Créer des routines supplémentaires pour gérer les ensembles de données ouverts.

## Considérations concernant la migration

Tenez compte des éléments suivants lors de la migration d'un schéma SQL Server vers MySQL :

- MySQL ne supporte pas MERGE cette déclaration. Cependant, AWS SCT vous pouvez émuler l'MERGE instruction lors de la conversion en utilisant la INSERT ON DUPLICATE KEY clause et les UPDATE FROM and DELETE FROM instructions.

Pour une émulation correcte INSERT ON DUPLICATE KEY, assurez-vous qu'une contrainte ou une clé primaire unique existe sur la base de données MySQL cible.

- Vous pouvez utiliser une GOTO instruction et une étiquette pour modifier l'ordre dans lequel les instructions sont exécutées. Toutes les instructions Transact-SQL qui suivent une GOTO instruction sont ignorées et le traitement se poursuit au niveau de l'étiquette. Vous pouvez utiliser GOTO des

instructions et des étiquettes n'importe où dans une procédure, un lot ou un bloc d'instructions. Vous pouvez également imbriquer GOTO des relevés.

MySQL n'utilise pas d'GOTOinstructions. Lors de la AWS SCT conversion du code contenant une GOTO instruction, il convertit l'instruction pour utiliser une LOOP...END LOOP instruction BEGIN...END or. Vous trouverez des exemples de AWS SCT conversion d'GOTOinstructions dans le tableau suivant.

Instruction SQL Server	Instruction MySQL
<pre>BEGIN   ....   statement1;   ....   GOTO label1;   statement2;   ....   label1:   Statement3;   .... END</pre>	<pre>BEGIN   label1:   BEGIN     ....     statement1;     ....     LEAVE label1;     statement2;     ....   END;   Statement3;   .... END</pre>

Instruction SQL Server	Instruction MySQL
<pre>BEGIN   ....   statement1;   ....   label1:   statement2;   ....   GOTO label1;   statement3;   ....   statement4;   .... END</pre>	<pre>BEGIN   ....   statement1;   ....   label1:   LOOP     statement2;     ....     ITERATE label1;   LEAVE label1; END LOOP;   statement3;   ....   statement4;   .... END</pre>
<pre>BEGIN   ....   statement1;   ....   label1:   statement2;   ....   statement3;   ....   statement4;   .... END</pre>	<pre>BEGIN   ....   statement1;   ....   label1:   BEGIN     statement2;     ....     statement3;     ....     statement4;   ....   END; END</pre>

- MySQL ne prend pas en charge les fonctions multiinstructions à valeur de table. AWS SCT simule des fonctions basées sur des tables lors d'une conversion en créant des tables temporaires et en réécrivant des instructions pour utiliser ces tables temporaires.

# Migration de SQL Server vers PostgreSQL avec AWS Schema Conversion Tool

Vous pouvez utiliser le pack d'extension SQL Server vers PostgreSQL dans AWS SCT. Ce pack d'extension émule les fonctions de base de données SQL Server dans le code PostgreSQL converti. Utilisez le pack d'extension SQL Server vers PostgreSQL pour émuler SQL Server Agent et SQL Server Database Mail. Pour plus d'informations sur les packs d'extension, consultez [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

## Rubriques

- [Privilèges pour PostgreSQL en tant que base de données cible](#)
- [Paramètres de conversion SQL Server vers PostgreSQL](#)
- [Conversion de partitions SQL Server en partitions PostgreSQL version 10](#)
- [Considérations concernant la migration](#)
- [Utilisation d'un pack d' AWS SCT extension pour émuler l'agent SQL Server dans PostgreSQL](#)
- [Utilisation d'un pack d' AWS SCT extension pour émuler SQL Server Database Mail dans PostgreSQL](#)

## Privilèges pour PostgreSQL en tant que base de données cible

Pour utiliser PostgreSQL comme cible AWS SCT, le privilège est requis. `CREATE ON DATABASE`. Assurez-vous d'accorder ce privilège à chaque base de données PostgreSQL cible.

Pour utiliser les synonymes publics convertis, remplacez le chemin de recherche par défaut de la base de données par `"$user", public_synonyms, public`.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les privilèges.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Dans PostgreSQL, seul le propriétaire du schéma ou un `superuser` peut supprimer un schéma. Le propriétaire peut supprimer un schéma et tous les objets qu'il inclut même si le propriétaire du schéma ne possède pas certains de ses objets.

Lorsque vous utilisez différents utilisateurs pour convertir et appliquer différents schémas à votre base de données cible, un message d'erreur peut s'afficher lorsque vous ne pouvez pas supprimer un schéma. Pour éviter ce message d'erreur, utilisez le rôle `superuser`.

## Paramètres de conversion SQL Server vers PostgreSQL

Pour modifier les paramètres de conversion de SQL Server vers PostgreSQL, choisissez Paramètres, puis Paramètres de conversion. Dans la liste supérieure, choisissez SQL Server, puis SQL Server — PostgreSQL. AWS SCT affiche tous les paramètres disponibles pour la conversion de SQL Server vers PostgreSQL.

Les paramètres de conversion de SQL Server vers PostgreSQL incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Permettre d'utiliser des index portant le même nom dans différentes tables de SQL Server.

Dans PostgreSQL, tous les noms d'index que vous utilisez dans le schéma doivent être uniques. Pour vous assurer que cela AWS SCT génère des noms uniques pour tous vos index, sélectionnez Générer des noms uniques pour les index.

- Pour convertir les procédures SQL Server en fonctions PostgreSQL.

Les versions 10 et antérieures de PostgreSQL ne prennent pas en charge les procédures. Les clients qui ne sont pas habitués à utiliser les procédures dans PostgreSQL peuvent les convertir AWS SCT en fonctions. Pour ce faire, sélectionnez Convertir les procédures en fonctions.

- Pour émuler la sortie de EXEC dans un tableau.

Votre base de données SQL Server source peut stocker le résultat de EXEC dans une table. AWS SCT crée des tables temporaires et une procédure supplémentaire pour émuler cette fonctionnalité. Pour utiliser cette émulation, sélectionnez Créer des routines supplémentaires pour gérer les ensembles de données ouverts.

- Pour définir le modèle à utiliser pour les noms de schéma dans le code converti. Pour le modèle de génération de nom de schéma, choisissez l'une des options suivantes :
  - <source\_db>— Utilise le nom de base de données SQL Server comme nom de schéma dans PostgreSQL.
  - <source\_schema>— Utilise le nom du schéma SQL Server comme nom de schéma dans PostgreSQL.
  - \_ <source\_db><schema>— Utilise une combinaison de noms de schéma et de base de données SQL Server comme nom de schéma dans PostgreSQL.
- Pour conserver les noms de vos objets source en majuscules.

Pour éviter de convertir les noms d'objets en minuscules, sélectionnez Éviter de convertir les noms en minuscules pour les opérations faisant la distinction entre majuscules et minuscules. Cette option s'applique uniquement lorsque vous activez la distinction majuscules/minuscules dans votre base de données cible.

- Pour conserver les noms des paramètres dans votre base de données source.

Pour ajouter des guillemets doubles aux noms des paramètres dans le code converti, sélectionnez Conserver les noms des paramètres d'origine.

## Conversion de partitions SQL Server en partitions PostgreSQL version 10

Lorsque vous convertissez une base de données Microsoft SQL Server vers Amazon Aurora PostgreSQL Compatible Edition (Aurora PostgreSQL) ou Amazon Relational Database Service for PostgreSQL (Amazon RDS for PostgreSQL), tenez compte des points suivants.

Dans SQL Server, vous créez des partitions avec des fonctions de partition. Lorsque vous convertissez une table SQL Server divisée en portions en une table PostgreSQL version 10 divisée en portions, vous devez être conscient de plusieurs problèmes potentiels :

- SQL Server vous permet de partitionner une table à l'aide d'une colonne sans contrainte NOT NULL. Dans ce cas, toutes les valeurs NULL sont dirigées vers la partition la plus à gauche. PostgreSQL ne prend pas en charge les valeurs NULL pour le partitionnement RANGE.

- SQL Server vous permet de créer des clés primaires et uniques pour les tables partitionnées. Pour PostgreSQL, vous créez des clés primaires ou uniques pour chaque partition directement. Par conséquent, la contrainte PRIMARY ou UNIQUE KEY doit être supprimée de la table parent lors de la migration vers PostgreSQL. Les noms de clé qui en résultent prennent le format `<original_key_name>_<partition_number>`.
- SQL Server vous permet de créer une contrainte de clé étrangère à partir de et vers des tables partitionnées. PostgreSQL ne prend pas en charge les clés étrangères qui référencent des tables partitionnées. De plus, PostgreSQL ne prend pas en charge les références de clé étrangère à partir d'une table partitionnée vers une autre table.
- SQL Server vous permet de créer des index pour les tables partitionnées. Pour PostgreSQL, un index doit être créé pour chaque partition directement. Par conséquent, les index doivent être supprimés des tables parents lors de la migration vers PostgreSQL. Les noms d'index qui en résultent sont au format `<original_index_name>_<partition_number>`.
- PostgreSQL ne prend pas en charge les index partitionnés.

## Considérations concernant la migration

Voici quelques points à prendre en compte lors de la migration d'un schéma SQL Server vers PostgreSQL :

- Dans PostgreSQL, tous les noms des objets dans un schéma doivent être uniques, y compris les index. Les noms d'index doivent être uniques dans le schéma de la table de base. Dans SQL Server, un nom d'index peut être identique pour différentes tables.

Pour garantir l'unicité des noms d'index, vous AWS SCT donne la possibilité de générer des noms d'index uniques si vos noms d'index ne sont pas uniques. Pour cela, choisissez l'option `Generate unique index names` (Générer des noms d'index uniques) dans les propriétés du projet. Cette option est activée par défaut. Si cette option est activée, les noms d'index uniques sont créés au format `IX_table_name_index_name`. Si cette option est désactivée, les noms d'index ne sont pas modifiés.

- Une instruction GOTO et une étiquette peuvent être utilisées pour modifier l'ordre dans lequel les instructions sont exécutées. Toutes les instructions Transact-SQL qui suivent une instruction GOTO sont ignorées et le traitement continue au niveau de l'étiquette. Les instructions GOTO et les étiquettes peuvent être utilisées à n'importe quel endroit dans une procédure, un lot ou un bloc d'instructions. Les instructions GOTO peuvent également être imbriquées.

PostgreSQL n'utilise pas les instructions GOTO. Lors de la AWS SCT conversion du code contenant une instruction GOTO, il convertit l'instruction pour utiliser une instruction BEGIN... END ou LOOP... END LOOP. Vous trouverez des exemples de AWS SCT conversion des instructions GOTO dans le tableau suivant.

### Instructions GOTO SQL Server et instructions PostgreSQL converties

Instruction SQL Server	Instruction PostgreSQL
<pre>BEGIN   ....   statement1;   ....   GOTO label1;   statement2;   ....   label1:   Statement3;   .... END</pre>	<pre>BEGIN   label1:   BEGIN     ....     statement1;     ....     EXIT label1;     statement2;     ....   END;   Statement3;   .... END</pre>
<pre>BEGIN   ....   statement1;   ....   label1:   statement2;   ....   GOTO label1;   statement3;   ....   statement4;   .... END</pre>	<pre>BEGIN   ....   statement1;   ....   label1:   LOOP     statement2;     ....     CONTINUE label1;     EXIT label1;   END LOOP;   statement3;   ....   statement4;   .... END</pre>



## Instruction SQL Server

```
BEGIN
  ....
  statement1;
  ....
  label1:
  statement2;
  ....
  statement3;
  ....
  statement4;
  ....
END
```

## Instruction PostgreSQL

```
BEGIN
  ....
  statement1;
  ....
  label1:
  BEGIN
    statement2;
    ....
    statement3;
    ....
    statement4;
    ....
  END;
END
```

- PostgreSQL ne prend pas en charge les instructions MERGE. AWS SCT émule le comportement d'une instruction MERGE de la manière suivante :
  - Par une construction INSERT ON CONFLICT.
  - En utilisant l'instruction UPDATE FROM DML, par exemple MERGE sans clause WHEN NOT MATCHED.
  - En utilisant une instruction CURSOR, par exemple avec une clause MERGE avec DELETE ou à l'aide d'une instruction de condition MERGE ON complexe.
- AWS SCT peut ajouter des déclencheurs de base de données à l'arborescence d'objets lorsque Amazon RDS est la cible.
- AWS SCT peut ajouter des déclencheurs au niveau du serveur à l'arborescence des objets lorsque Amazon RDS est la cible.
- SQL Server crée et gère automatiquement des tables `deleted` et `inserted`. Vous pouvez utiliser ces tables temporaires résidant en mémoire pour tester les effets de certaines modifications de données et pour définir les conditions des actions de déclenchement DML. AWS SCT peut convertir l'utilisation de ces tables dans des instructions de déclenchement DML.
- AWS SCT peut ajouter des serveurs liés à l'arborescence des objets lorsque Amazon RDS est la cible.
- Lors de la migration de Microsoft SQL Server vers PostgreSQL, la fonction `SUSER_SNAME` intégrée est convertie comme suit :

- `SUSER_SNAME` – Renvoie le nom de connexion associé à un numéro d'identification de sécurité (SID).
- `SUSER_SNAME(<server_user_sid>)` – Non pris en charge.
- `SUSER_SNAME () CURRENT_USER` – Renvoie le nom d'utilisateur du contexte d'exécution actuel.
- `SUSER_SNAME (NULL)` - Renvoie NULL.
- La conversion de fonctions avec valeurs de table est prise en charge. Les fonctions avec valeurs de table renvoient une table et peuvent remplacer une table dans une requête.
- `PATINDEX` renvoie la position de départ de la première occurrence d'un modèle dans une expression spécifiée sur tous les types de données texte et caractère. Il renvoie des zéros si le modèle n'est pas trouvé. `<pattern character><expression character varying>`Lors de la conversion de SQL Server vers Amazon RDS for AWS SCT PostgreSQL, le code d'application qui utilise `PATINDEX` est remplacé par `aws_sqlserver_ext.patindex (,)`.
- Dans SQL Server, un type de table défini par l'utilisateur est un type qui représente la définition d'une structure de table. Vous utilisez un type de table défini par l'utilisateur afin de déclarer les paramètres de valeur de table pour les procédures ou fonctions stockées. Vous pouvez également utiliser un type de table défini par l'utilisateur pour déclarer les variables de table que vous souhaitez utiliser dans un lot ou dans le corps d'une procédure stockée ou d'une fonction. AWS SCT a émulé ce type dans PostgreSQL en créant une table temporaire.

Lors de la conversion de SQL Server vers PostgreSQL AWS SCT , les objets système SQL Server sont convertis en objets reconnaissables dans PostgreSQL. Le tableau suivant montre la façon dont les objets système sont convertis.

Cas d'utilisation de MS SQL Server	Substitution par PostgreSQL
<code>SYS.SCHEMAS</code>	<code>AWS_SQLSERVER_EXT.SYS_SCHEMAS</code>
<code>SYS.TABLES</code>	<code>AWS_SQLSERVER_EXT.SYS_TABLES</code>
<code>SYS.VIEWS</code>	<code>AWS_SQLSERVER_EXT.SYS_VIEWS</code>
<code>SYS.ALL_VIEWS</code>	<code>AWS_SQLSERVER_EXT.SYS_ALL_VIEWS</code>

Cas d'utilisation de MS SQL Server	Substitution par PostgreSQL
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVERCLÉS ÉTRANGÈRES_EXT.SYS_SYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEX
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVEROBJETS_EXT.SYS_SYS
SYS.SQL_MODULES	AWS_SQLSERVERMODULES_EXT.SYS_SQL
SYS.DATABASES	AWS_SQLSERVERBASES DE DONNÉES_EXT.SYS_
INFORMATION_SCHEMA.SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHÉMAS

Cas d'utilisation de MS SQL Server	Substitution par PostgreSQL
INFORMATION_SCHEMA.VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA.TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA.COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA.CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA.TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS
INFORMATION_SCHEMA.KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLUMN_USAGE
INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_TABLE_USAGE
INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_COLUMN_USAGE
INFORMATION_SCHEMA.ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROUTINES
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROCESSES
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

## Utilisation d'un pack d' AWS SCT extension pour émuler l'agent SQL Server dans PostgreSQL

SQL Server Agent est un service Microsoft Windows qui exécute des tâches SQL Server. L'agent SQL Server exécute les tâches selon un calendrier, en réponse à un événement spécifique ou à la demande. Pour plus d'informations sur SQL Server Agent, consultez la [documentation technique Microsoft](#).

PostgreSQL n'a pas d'équivalent pour SQL Server Agent. Pour émuler les fonctionnalités de l'agent SQL Server, AWS SCT crée un pack d'extension. Ce pack d'extension utilise AWS Lambda Amazon CloudWatch. AWS Lambda implémente l'interface que vous utilisez pour gérer les plannings et exécuter des tâches. Amazon CloudWatch applique les règles de planification.

AWS Lambda et Amazon CloudWatch utilisent un paramètre JSON pour interagir. La structure de ce paramètre JSON est la suivante.

```
{
  "mode": mode,
  "parameters": {
    list of parameters
  },
  "callback": procedure name
}
```

Dans l'exemple précédent, *mode* il s'agit du type de tâche et *list of parameters* d'un ensemble de paramètres qui dépendent du type de tâche. *procedure name* C'est également le nom de la procédure qui s'exécute une fois la tâche terminée.

AWS SCT utilise une fonction Lambda pour contrôler et exécuter des tâches. La CloudWatch règle lance l'exécution de la tâche et fournit les informations nécessaires pour démarrer la tâche. Lorsque la CloudWatch règle se déclenche, elle lance la fonction Lambda en utilisant les paramètres de la règle.

Pour créer une tâche simple qui appelle une procédure, utilisez le format suivant.

```
{
  "mode": "run_job",
  "parameters": {
    "vendor": "mysql",
    "cmd": "lambda_db.nightly_job"
  }
}
```

```
}  
}
```

Pour créer une tâche en plusieurs étapes, utilisez le format suivant.

```
{  
  "mode": "run_job",  
  "parameters": {  
    "job_name": "Job1",  
    "enabled": "true",  
    "start_step_id": 1,  
    "notify_level_email": [0|1|2|3],  
    "notify_email": email,  
    "delete_level": [0|1|2|3],  
    "job_callback": "ProcCallBackJob(job_name, code, message)",  
    "step_callback": "ProcCallBackStep(job_name, step_id, code, message)"  
  },  
  "steps": [  
    {  
      "id":1,  
      "cmd": "ProcStep1",  
      "cmdexec_success_code": 0,  
      "on_success_action": [2|3|4],  
      "on_success_step_id": 1,  
      "on_fail_action": 0,  
      "on_fail_step_id": 0,  
      "retry_attempts": number,  
      "retry_interval": number  
    },  
    {  
      "id":2,  
      "cmd": "ProcStep2",  
      "cmdexec_success_code": 0,  
      "on_success_action": [1|2|3|4],  
      "on_success_step_id": 0,  
      "on_fail_action": 0,  
      "on_fail_step_id": 0,  
      "retry_attempts": number,  
      "retry_interval": number  
    },  
    ...  
  ]  
}
```

Pour émuler le comportement de l'agent SQL Server dans PostgreSQL, AWS SCT le pack d'extension crée également les tables et procédures suivantes.

Tables qui émulent l'agent SQL Server dans PostgreSQL

Pour émuler l'agent SQL Server, le pack d'extension utilise les tableaux suivants :

sysjobs

Stocke les informations relatives aux tâches.

étapes du sysjob

Stocke les informations relatives aux étapes d'une tâche.

plannings du système

Stocke les informations relatives aux plannings de travail.

plannings de tâches sysjob

Stocke les informations de planification pour les tâches individuelles.

sysjobhistory

Stocke les informations relatives aux exécutions des tâches planifiées.

Procédures qui émulent l'agent SQL Server dans PostgreSQL

Pour émuler l'agent SQL Server, le pack d'extension utilise les procédures suivantes :

sp\_add\_job

Ajoute une nouvelle tâche.

sp\_add\_jobstep

Ajoute une étape à une tâche.

sp\_add\_schedule

Crée une nouvelle règle de planification dans Amazon CloudWatch. Vous pouvez utiliser ce calendrier avec autant de tâches que vous le souhaitez.

sp\_attach\_schedule

Définit un calendrier pour le travail sélectionné.

## sp\_add\_jobschedule

Crée une règle de planification pour une tâche sur Amazon CloudWatch et définit l'objectif de cette règle.

## sp\_update\_job

Met à jour les attributs de la tâche créée précédemment.

## sp\_update\_jobstep

Met à jour les attributs de l'étape d'une tâche.

## sp\_update\_schedule

Met à jour les attributs d'une règle de planification dans Amazon CloudWatch.

## sp\_update\_jobschedule

Met à jour les attributs du planning pour la tâche spécifiée.

## sp\_delete\_job

Supprime une tâche.

## sp\_delete\_jobstep

Supprime une étape d'une tâche.

## sp\_delete\_planning

Supprime un planning.

## sp\_delete\_jobschedule

Supprime d'Amazon CloudWatch la règle de planification pour la tâche spécifiée.

## sp\_detach\_schedule

Supprime une association entre un planning et une tâche.

## get\_jobs, update\_job

Procédures internes qui interagissent avec AWS Elastic Beanstalk.

sp\_verify\_job\_date, sp\_verify\_job\_time, sp\_verify\_job, sp\_verify\_jobstep, sp\_verify\_schedule, sp\_verify\_job\_identifiers, sp\_verify\_schedule\_identifiers

Procédures internes qui vérifient les paramètres.



## Syntaxe des procédures qui émulent l'agent SQL Server dans PostgreSQL

La `aws_sqlserver_ext.sp_add_job` procédure du pack d'extension émule la `msdb.dbo.sp_add_job` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_name varchar,  
par_enabled smallint = 1,  
par_description varchar = NULL::character varying,  
par_start_step_id integer = 1,  
par_category_name varchar = NULL::character varying,  
par_category_id integer = NULL::integer,  
par_owner_login_name varchar = NULL::character varying,  
par_notify_level_eventlog integer = 2,  
par_notify_level_email integer = 0,  
par_notify_level_netsend integer = 0,  
par_notify_level_page integer = 0,  
par_notify_email_operator_name varchar = NULL::character varying,  
par_notify_netsend_operator_name varchar = NULL::character varying,  
par_notify_page_operator_name varchar = NULL::character varying,  
par_delete_level integer = 0,  
inout par_job_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

La `aws_sqlserver_ext.sp_add_jobstep` procédure du pack d'extension émule la `msdb.dbo.sp_add_jobstep` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
par_step_name varchar = NULL::character varying,  
par_subsystem varchar = 'TSQL'::bpchar,  
par_command text = NULL::text,  
par_additional_parameters text = NULL::text,  
par_cmdexec_success_code integer = 0,  
par_on_success_action smallint = 1,  
par_on_success_step_id integer = 0,  
par_on_fail_action smallint = 2,  
par_on_fail_step_id integer = 0,  
par_server varchar = NULL::character varying,
```

```
par_database_name varchar = NULL::character varying,  
par_database_user_name varchar = NULL::character varying,  
par_retry_attempts integer = 0,  
par_retry_interval integer = 0,  
par_os_run_priority integer = 0,  
par_output_file_name varchar = NULL::character varying,  
par_flags integer = 0,  
par_proxy_id integer = NULL::integer,  
par_proxy_name varchar = NULL::character varying,  
inout par_step_uid char = NULL::bpchar,  
out returncode integer
```

La `aws_sqlserver_ext.sp_add_schedule` procédure du pack d'extension émule la `msdb.dbo.sp_add_schedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_schedule_name varchar,  
par_enabled smallint = 1,  
par_freq_type integer = 0,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,  
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
par_owner_login_name varchar = NULL::character varying,  
*inout par_schedule_uid char = NULL::bpchar,*  
inout par_schedule_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

La `aws_sqlserver_ext.sp_attach_schedule` procédure du pack d'extension émule la `msdb.dbo.sp_attach_schedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,
```

```
par_automatic_post smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sp_add_jobschedule` procédure du pack d'extension émule la `msdb.dbo.sp_add_jobschedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_enabled smallint = 1,  
par_freq_type integer = 1,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,  
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
inout par_schedule_id integer = NULL::integer,  
par_automatic_post smallint = 1,  
inout par_schedule_uid char = NULL::bpchar,  
out returncode integer
```

La `aws_sqlserver_ext.sp_delete_job` procédure du pack d'extension émule la `msdb.dbo.sp_delete_job` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_originating_server varchar = NULL::character varying,  
par_delete_history smallint = 1,  
par_delete_unused_schedule smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sp_delete_jobstep` procédure du pack d'extension émule la `msdb.dbo.sp_delete_jobstep` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
out returncode integer
```

La `aws_sqlserver_ext.sp_delete_jobschedule` procédure du pack d'extension émule la `msdb.dbo.sp_delete_jobschedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_keep_schedule integer = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sp_delete_schedule` procédure du pack d'extension émule la `msdb.dbo.sp_delete_schedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_force_delete smallint = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sp_detach_schedule` procédure du pack d'extension émule la `msdb.dbo.sp_detach_schedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_delete_unused_schedule smallint = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sp_update_job` procédure du pack d'extension émule la `msdb.dbo.sp_update_job` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_description varchar = NULL::character varying
par_start_step_id integer = NULL::integer
par_category_name varchar = NULL::character varying
par_owner_login_name varchar = NULL::character varying
par_notify_level_eventlog integer = NULL::integer
par_notify_level_email integer = NULL::integer
par_notify_level_netsend integer = NULL::integer
par_notify_level_page integer = NULL::integer
par_notify_email_operator_name varchar = NULL::character varying
par_notify_netsend_operator_name varchar = NULL::character varying
par_notify_page_operator_name varchar = NULL::character varying
par_delete_level integer = NULL::integer
par_automatic_post smallint = 1
out returncode integer
```

La `aws_sqlserver_ext.sp_update_jobschedule` procédure du pack d'extension émule la `msdb.dbo.sp_update_jobschedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
par_active_end_time integer = NULL::integer
```

```
par_automatic_post smallint = 1
out_returncode integer
```

La `aws_sqlserver_ext.sp_update_jobstep` procédure du pack d'extension émule la `msdb.dbo.sp_update_jobstep` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_step_id integer = NULL::integer
par_step_name varchar = NULL::character varying
par_subsystem varchar = NULL::character varying
par_command text = NULL::text
par_additional_parameters text = NULL::text
par_cmdexec_success_code integer = NULL::integer
par_on_success_action smallint = NULL::smallint
par_on_success_step_id integer = NULL::integer
par_on_fail_action smallint = NULL::smallint
par_on_fail_step_id integer = NULL::integer
par_server varchar = NULL::character varying
par_database_name varchar = NULL::character varying
par_database_user_name varchar = NULL::character varying
par_retry_attempts integer = NULL::integer
par_retry_interval integer = NULL::integer
par_os_run_priority integer = NULL::integer
par_output_file_name varchar = NULL::character varying
par_flags integer = NULL::integer
par_proxy_id integer = NULL::integer
par_proxy_name varchar = NULL::character varying
out_returncode integer
```

La `aws_sqlserver_ext.sp_update_schedule` procédure du pack d'extension émule la `msdb.dbo.sp_update_schedule` procédure. Pour plus d'informations sur la procédure source de l'agent SQL Server, consultez [la documentation technique Microsoft](#).

```
par_schedule_id integer = NULL::integer
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
```

```
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
par_active_end_time integer = NULL::integer
par_owner_login_name varchar = NULL::character varying
par_automatic_post smallint = 1
out returncode integer
```

## Exemples d'utilisation de procédures qui émulent l'agent SQL Server dans PostgreSQL

Pour ajouter une nouvelle tâche, suivez la `aws_sqlserver_ext.sp_add_job` procédure ci-dessous.

```
SELECT * FROM aws_sqlserver_ext.sp_add_job (
    par_job_name := 'test_job',
    par_enabled := 1::smallint,
    par_start_step_id := 1::integer,
    par_category_name := '[Uncategorized (Local)]',
    par_owner_login_name := 'sa');
```

Pour ajouter une nouvelle étape de travail, suivez la `aws_sqlserver_ext.sp_add_jobstep` procédure décrite ci-dessous.

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobstep (
    par_job_name := 'test_job',
    par_step_id := 1::smallint,
    par_step_name := 'test_job_step1',
    par_subsystem := 'TSQL',
    par_command := 'EXECUTE [dbo].[PROC_TEST_JOB_STEP1];',
    par_server := NULL,
    par_database_name := 'GOLD_TEST_SS');
```

Pour ajouter un calendrier simple, suivez la `aws_sqlserver_ext.sp_add_schedule` procédure ci-dessous.

```
SELECT * FROM aws_sqlserver_ext.sp_add_schedule(
    par_schedule_name := 'RunOnce',
    par_freq_type := 1,
```

```
par_active_start_time := 233000);
```

Pour définir un calendrier pour une tâche, suivez la `aws_sqlserver_ext.sp_attach_schedule` procédure décrite ci-dessous.

```
SELECT * FROM aws_sqlserver_ext.sp_attach_schedule (  
    par_job_name := 'test_job',  
    par_schedule_name := 'NightlyJobs');
```

Pour créer un calendrier pour une tâche, suivez la `aws_sqlserver_ext.sp_add_jobschedule` procédure décrite ci-dessous.

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobschedule (  
    par_job_name := 'test_job2',  
    par_name := 'test_schedule2',  
    par_enabled := 1::smallint,  
    par_freq_type := 4,  
    par_freq_interval := 1,  
    par_freq_subday_type := 4,  
    par_freq_subday_interval := 1,  
    par_freq_relative_interval := 0,  
    par_freq_recurrence_factor := 0,  
    par_active_start_date := 20100801,  
    par_active_end_date := 99991231,  
    par_active_start_time := 0,  
    par_active_end_time := 0);
```

## Exemples d'utilisation pour émuler l'agent SQL Server dans PostgreSQL

Si le code de votre base de données source utilise l'agent SQL Server pour exécuter des tâches, vous pouvez utiliser le pack d'extension SQL Server vers PostgreSQL AWS SCT pour convertir ce code en PostgreSQL. Le pack d'extension utilise des AWS Lambda fonctions pour émuler le comportement de SQL Server Agent.

Vous pouvez créer une nouvelle AWS Lambda fonction ou enregistrer une fonction existante.

### Pour créer une nouvelle AWS Lambda fonction

1. Dans AWS SCT l'arborescence de la base de données cible, ouvrez le menu contextuel (clic droit), choisissez Appliquer le pack d'extension pour, puis PostgreSQL.



L'assistant Kit d'extension s'affiche.

2. Dans l'onglet Service d'émulation de l'agent SQL Server, procédez comme suit :
  - Choisissez Créer une AWS Lambda fonction.
  - Pour Connexion à la base de données, entrez le nom de l'utilisateur de la base de données cible.
  - Dans le champ Mot de passe de la base de données, entrez le mot de passe correspondant au nom d'utilisateur que vous avez saisi à l'étape précédente.
  - Pour le dossier de bibliothèque Python, entrez le chemin d'accès au dossier de votre bibliothèque Python.
  - Choisissez Créer AWS Lambda une fonction, puis Next.

Pour enregistrer une AWS Lambda fonction que vous avez déployée précédemment

- Exécutez le script suivant sur votre base de données cible.

```
SELECT
  FROM aws_sqlserver_ext.set_service_setting(
    p_service := 'JOB',
    p_setting := 'LAMBDA_ARN',
    p_value := ARN)
```

Dans l'exemple précédent, *ARN* il s'agit de l'Amazon Resource Name (ARN) de la AWS Lambda fonction déployée.

L'exemple suivant crée une tâche simple composée d'une étape. Toutes les cinq minutes, cette tâche exécute la `job_example` fonction créée précédemment. Cette fonction insère des enregistrements dans la `job_example_table` table.

Pour créer cette tâche simple

1. Créez une tâche à l'aide de la `aws_sqlserver_ext.sp_add_job` fonction illustrée ci-dessous.

```
SELECT
  FROM aws_sqlserver_ext.sp_add_job (
```

```
par_job_name := 'test_simple_job');
```

2. Créez une étape de travail à l'aide de la `aws_sqlserver_ext.sp_add_jobstep` fonction illustrée ci-dessous.

```
SELECT
FROM aws_sqlserver_ext.sp_add_jobstep (
    par_job_name := 'test_simple_job',
    par_step_name := 'test_simple_job_step1',
    par_command := 'PERFORM job_simple_example;');
```

L'étape de travail indique le rôle de la fonction.

3. Créez un planificateur pour le travail à l'aide de la `aws_sqlserver_ext.sp_add_jobschedule` fonction illustrée ci-dessous.

```
SELECT
FROM aws_sqlserver_ext.sp_add_jobschedule (
    par_job_name := 'test_simple_job',
    par_name := 'test_schedule',
    par_freq_type := 4, /* Daily */
    par_freq_interval := 1, /* frequency_interval is unused */
    par_freq_subday_type := 4, /* Minutes */
    par_freq_subday_interval := 5 /* 5 minutes */);
```

L'étape de travail indique le rôle de la fonction.

Pour supprimer cette tâche, utilisez la `aws_sqlserver_ext.sp_delete_job` fonction illustrée ci-dessous.

```
PERFORM aws_sqlserver_ext.sp_delete_job(
    par_job_name := 'PeriodicJob1'::character varying,
    par_delete_history := 1::smallint,
    par_delete_unused_schedule := 1::smallint);
```

## Utilisation d'un pack d' AWS SCT extension pour émuler SQL Server Database Mail dans PostgreSQL

Vous pouvez utiliser SQL Server Database Mail pour envoyer des e-mails aux utilisateurs à partir du moteur de base de données SQL Server ou d'une instance gérée Azure SQL. Ces e-mails peuvent

contenir les résultats de requêtes ou inclure des fichiers provenant de n'importe quelle ressource de votre réseau. Pour plus d'informations sur SQL Server Database Mail, consultez [la documentation technique Microsoft](#).

PostgreSQL n'a pas d'équivalent pour SQL Server Database Mail. Pour émuler les fonctionnalités de messagerie de base de données SQL Server, AWS SCT crée un pack d'extension. Ce pack d'extension utilise AWS Lambda Amazon Simple Email Service (Amazon SES). AWS Lambda fournit aux utilisateurs une interface leur permettant d'interagir avec le service d'envoi d'e-mails Amazon SES. Pour configurer cette interaction, ajoutez le nom de ressource Amazon (ARN) de votre fonction Lambda.

Pour créer un nouveau compte de messagerie, utilisez la commande suivante.

```
do
$$
begin
PERFORM sysmail_add_account_sp (
    par_account_name := 'your_account_name',
    par_email_address := 'your_account_email',
    par_display_name := 'your_account_display_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

Pour ajouter l'ARN de votre fonction Lambda au compte de messagerie existant, utilisez la commande suivante.

```
do
$$
begin
PERFORM sysmail_update_account_sp (
    par_account_name := 'existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

Dans les exemples précédents, *ARN* c'est l'ARN de votre fonction Lambda.

Pour émuler le comportement du courrier de base de données SQL Server dans PostgreSQL, AWS SCT le pack d'extension utilise les tables, les vues et les procédures suivantes.

Tables qui émulent le courrier de base de données SQL Server dans PostgreSQL

Pour émuler SQL Server Database Mail, le pack d'extension utilise les tables suivantes :

compte sysmail\_

Stocke les informations relatives aux comptes de messagerie.

sysmail\_profile

Stocke les informations relatives aux profils utilisateur.

serveur\_système

Stocke les informations relatives aux serveurs de messagerie.

sysmail\_mailitems

Stocke la liste des e-mails.

sysmail\_attachments

Contient une ligne pour chaque pièce jointe à un e-mail.

journal de messagerie système

Stocke les informations de service relatives à l'envoi de messages électroniques.

compte sysmail\_profile

Stocke les informations relatives aux profils utilisateurs et aux comptes de messagerie.

Vues qui émulent le courrier de base de données SQL Server dans PostgreSQL

Pour émuler SQL Server Database Mail, AWS SCT créez les vues suivantes dans la base de données PostgreSQL afin de garantir la compatibilité. Le pack d'extension ne les utilise pas, mais votre code converti peut interroger ces vues.

sysmail\_allitems

Inclut une liste de tous les e-mails.

## sysmail\_faileditems

Inclut une liste des e-mails qui n'ont pas pu être envoyés.

## sysmail\_sentitems

Inclut une liste des e-mails envoyés.

## sysmail\_unsentitems

Inclut une liste d'e-mails qui n'ont pas encore été envoyés.

## sysmail\_mailattachments

Inclut une liste des fichiers joints.

## Procédures qui émulent le courrier de base de données SQL Server dans PostgreSQL

Pour émuler SQL Server Database Mail, le pack d'extension utilise les procédures suivantes :

### sp\_send\_dbmail

Envoie un e-mail aux destinataires spécifiés.

### sysmail\_add\_profile\_sp

Crée un nouveau profil utilisateur.

### sysmail\_add\_account\_sp

Crée un nouveau compte de messagerie qui stocke des informations telles que les informations d'identification du protocole SMTP (Simple Mail Transfer Protocol), etc.

### sysmail\_add\_profileaccount\_sp

Ajoute un compte e-mail au profil utilisateur spécifié.

### sysmail\_update\_profile\_sp

Modifie les attributs du profil utilisateur tels que la description, le nom, etc.

### sysmail\_update\_account\_sp

Modifie les informations du compte de messagerie existant.

### sysmail\_update\_profileaccount\_sp

Met à jour les informations du compte e-mail dans le profil utilisateur spécifié.

## sysmail\_delete\_profileaccount\_sp

Supprime un compte e-mail du profil utilisateur spécifié.

## sysmail\_delete\_account\_sp

Supprime le compte de messagerie.

## sysmail\_delete\_profile\_sp

Supprime le profil utilisateur.

## sysmail\_delete\_mailitems\_sp

Supprime les e-mails des tables internes.

## sysmail\_help\_profile\_sp

Affiche des informations sur le profil utilisateur.

## sysmail\_help\_account\_sp

Affiche les informations relatives au compte de messagerie.

## sysmail\_help\_profileaccount\_sp

Affiche des informations sur les comptes de messagerie associés au profil utilisateur.

## sysmail\_dbmail\_json

Procédure interne qui génère des requêtes JSON pour des AWS Lambda fonctions.

## sysmail\_verify\_profile\_sp, sysmail\_verify\_account\_sp, sysmail\_verify\_addressparams\_sp

Procédures internes qui vérifient les paramètres.

## sp\_get\_dbmail, sp\_set\_dbmail, sysmail\_dbmail\_xml

Procédures internes déconseillées.

Syntaxe des procédures qui émulent le courrier de base de données SQL Server dans PostgreSQL

La `aws_sqlserver_ext.sp_send_dbmail` procédure du pack d'extension émule la `msdb.dbo.sp_send_dbmail` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_name varchar = NULL::character varying,  
par_recipients text = NULL::text,  
par_copy_recipients text = NULL::text,  
par_blind_copy_recipients text = NULL::text,  
par_subject varchar = NULL::character varying,  
par_body text = NULL::text,  
par_body_format varchar = NULL::character varying,  
par_importance varchar = 'NORMAL'::character varying,  
par_sensitivity varchar = 'NORMAL'::character varying,  
par_file_attachments text = NULL::text,  
par_query text = NULL::text,  
par_execute_query_database varchar = NULL::character varying,  
par_attach_query_result_as_file smallint = 0,  
par_query_attachment_filename varchar = NULL::character varying,  
par_query_result_header smallint = 1,  
par_query_result_width integer = 256,  
par_query_result_separator VARCHAR = ' '::character varying,  
par_exclude_query_output smallint = 0,  
par_append_query_error smallint = 0,  
par_query_no_truncate smallint = 0,  
par_query_result_no_padding smallint = 0,  
out par_mailitem_id integer,  
par_from_address text = NULL::text,  
par_reply_to text = NULL::text,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_delete_mailitems_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_delete_mailitems_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,  
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_add_profile_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_add_profile_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,
```

```
out par_profile_id integer,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_add_account_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_add_account_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_account_name varchar  
par_email_address varchar  
par_display_name varchar = NULL::character varying  
par_replyto_address varchar = NULL::character varying  
par_description varchar = NULL::character varying  
par_mailserver_name varchar = NULL::character varying  
par_mailserver_type varchar = 'SMTP'::bpchar  
par_port integer = 25  
par_username varchar = NULL::character varying  
par_password varchar = NULL::character varying  
par_use_default_credentials smallint = 0  
par_enable_ssl smallint = 0  
out par_account_id integer  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_add_profileaccount_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_add_profileaccount_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_sequence_number integer = NULL::integer,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_help_profile_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_help_profile_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,
```



```
out returncode integer
```

La `aws_sqlserver_ext.sysmail_update_profile_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_update_profile_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_delete_profile_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_delete_profile_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_help_account_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_help_account_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

La `aws_sqlserver_ext.sysmail_update_account_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_update_account_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_email_address varchar = NULL::character varying,  
par_display_name varchar = NULL::character varying,  
par_replyto_address varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
par_mailserver_name varchar = NULL::character varying,
```

```
par_mailserver_type varchar = NULL::character varying,  
par_port integer = NULL::integer,  
par_username varchar = NULL::character varying,  
par_password varchar = NULL::character varying,  
par_use_default_credentials smallint = NULL::smallint,  
par_enable_ssl smallint = NULL::smallint,  
par_timeout integer = NULL::integer,  
par_no_credential_change smallint = NULL::smallint,  
out_returncode integer
```

La `aws_sqlserver_ext.sysmail_delete_account_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_delete_account_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out_returncode integer
```

La `aws_sqlserver_ext.sysmail_help_profileaccount_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_help_profileaccount_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out_returncode integer
```

La `aws_sqlserver_ext.sysmail_update_profileaccount_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_update_profileaccount_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_sequence_number integer = NULL::integer,  
out_returncode integer
```

La `aws_sqlserver_ext.sysmail_delete_profileaccount_sp` procédure du pack d'extension émule la `msdb.dbo.sysmail_delete_profileaccount_sp` procédure. Pour plus d'informations sur la procédure source de courrier de base de données SQL Server, consultez [la documentation technique Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out_returncode integer
```

Exemples d'utilisation de procédures qui émulent SQL Server Database Mail dans PostgreSQL

Pour envoyer un e-mail, suivez la `aws_sqlserver_ext.sp_send_dbmail` procédure ci-dessous.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',  
    par_body := 'The stored procedure finished'  
);
```

L'exemple suivant montre comment envoyer un e-mail contenant les résultats d'une requête.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Account with id = 1',  
    par_query := 'SELECT COUNT(*)FROM Account WHERE id = 1'  
);
```

L'exemple suivant montre comment envoyer un e-mail avec du code HTML.

```
DECLARE var_tableHTML TEXT;  
SET var_tableHTML := CONCAT(  
    '<H1>Work Order Report</H1>',  
    '<table border="1">',  
    '<tr><th>Work Order ID</th><th>Product ID</th>',  
    '<th>Name</th><th>Order Qty</th><th>Due Date</th>',  
    '<th>Expected Revenue</th></tr>',  
    '</table>'
```

```
);  
PERFORM sp_send_dbmail (  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Work Order List',  
    par_body := var_tableHTML,  
    par_body_format := 'HTML'  
);
```

Pour supprimer des e-mails, suivez la `aws_sqlserver_ext.sysmail_delete_mailitems_sp` procédure ci-dessous.

```
DECLARE var_GETDATE datetime;  
SET var_GETDATE = NOW();  
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_before := var_GETDATE  
);
```

L'exemple suivant montre comment supprimer les e-mails les plus anciens.

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_before := '31.12.2015'  
);
```

L'exemple suivant montre comment supprimer tous les e-mails qui ne peuvent pas être envoyés.

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_status := 'failed'  
);
```

Pour créer un nouveau profil utilisateur, suivez la `aws_sqlserver_ext.sysmail_add_profile_sp` procédure décrite ci-dessous.

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

L'exemple suivant montre comment créer un nouveau profil et enregistrer l'identifiant de profil unique dans une variable.

```
DECLARE var_profileId INT;
```

```
SELECT par_profile_id
  FROM sysmail_add_profile_sp (
    profile_name := 'Administrator',
    par_description := ' Profile used for administrative mail.')
 INTO var_profileId;

SELECT var_profileId;
```

Pour créer un nouveau compte de messagerie, suivez la `aws_sqlserver_ext.sysmail_add_account_sp` procédure ci-dessous.

```
PERFORM sysmail_add_account_sp (
  par_account_name := 'Audit Account',
  par_email_address := 'dba@rusgl.info',
  par_display_name := 'Test Automated Mailer',
  par_description := 'Account for administrative e-mail.',
  par_mailserver_type := 'AWSLAMBDA'
  par_mailserver_name := 'arn:aws:lambda:us-west-2:555555555555:function:pg_v3'
);
```

Pour ajouter un compte e-mail au profil utilisateur, suivez la `aws_sqlserver_ext.sysmail_add_profileaccount_sp` procédure ci-dessous.

```
PERFORM sysmail_add_profileaccount_sp (
  par_account_name := 'Administrator',
  par_account_name := 'Audit Account',
  par_sequence_number := 1
);
```

## Exemples d'utilisation pour émuler SQL Server Database Mail dans PostgreSQL

Si le code de votre base de données source utilise SQL Server Database Mail pour envoyer des e-mails, vous pouvez utiliser le pack d' AWS SCT extension pour convertir ce code en PostgreSQL.

Pour envoyer un e-mail depuis votre base de données PostgreSQL

1. Créez et configurez votre AWS Lambda fonction.
2. Appliquez le pack d' AWS SCT extension.
3. Créez un profil utilisateur à l'aide de la `sysmail_add_profile_sp` fonction illustrée ci-dessous.

4. Créez un compte e-mail à l'aide de la `sysmail_add_account_sp` fonction illustrée ci-dessous.
5. Ajoutez ce compte e-mail à votre profil utilisateur à l'aide de la `sysmail_add_profileaccount_sp` fonction illustrée ci-dessous.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.  
proc_dbmail_settings_msdb()  
RETURNS void  
AS  
$BODY$  
BEGIN  
PERFORM aws_sqlserver_ext.sysmail_add_profile_sp(  
    par_profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);  
PERFORM aws_sqlserver_ext.sysmail_add_account_sp(  
    par_account_name := 'Audit Account',  
    par_description := 'Account for administrative e-mail.',  
    par_email_address := 'dba@rusgl.info',  
    par_display_name := 'Test Automated Mailer',  
    par_mailserver_type := 'AWSLAMBDA',  
    par_mailserver_name := 'your_ARN'  
);  
PERFORM aws_sqlserver_ext.sysmail_add_profileaccount_sp(  
    par_profile_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);  
END;  
$BODY$  
LANGUAGE plpgsql;
```

6. Envoyez un e-mail à l'aide de la `sp_send_dbmail` fonction illustrée ci-dessous.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.  
proc_dbmail_send_msdb()  
RETURNS void  
AS  
$BODY$  
BEGIN  
PERFORM aws_sqlserver_ext.sp_send_dbmail(  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_body := 'The stored procedure finished',
```

```
    par_subject := 'Automated Success Message'  
);  
END;  
$BODY$  
LANGUAGE plpgsql;
```

Pour consulter les informations relatives à tous les profils utilisateur, suivez la `sysmail_help_profile_sp` procédure ci-dessous.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

L'exemple suivant affiche les informations relatives au profil utilisateur spécifique.

```
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_id := 1);  
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_name :=  
'Administrator');
```

Pour consulter les informations relatives à tous les comptes de messagerie, suivez la `sysmail_help_account_sp` procédure ci-dessous.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

L'exemple suivant affiche les informations relatives au compte de messagerie spécifique.

```
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_id := 1);  
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_name := 'Audit  
Account');
```

Pour consulter les informations relatives à tous les comptes de messagerie associés aux profils utilisateur, suivez la `sysmail_help_profileaccount_sp` procédure décrite ci-dessous.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

L'exemple suivant filtre les enregistrements par identifiant, nom de profil ou nom de compte.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1);  
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1,  
    par_account_id := 1);
```

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_name :=
  'Administrator');
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_account_name := 'Audit
  Account');
```

Pour modifier le nom ou la description du profil utilisateur, suivez la `sysmail_update_profile_sp` procédure ci-dessous.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
  par_profile_id := 2,
  par_profile_name := 'New profile name'
);
```

Pour modifier les paramètres du compte de messagerie, suivez la `ysmail_update_account_sp` procédure ci-dessous.

```
select from aws_sqlserver_ext.sysmail_update_account_sp (
  par_account_name := 'Audit Account',
  par_mailserver_name := 'arn:aws:lambda:region:XXXXXXXXXXXX:function:func_test',
  par_mailserver_type := 'AWSLAMBDA'
);
```

## Migration de SQL Server vers Amazon RDS for SQL Server avec AWS Schema Conversion Tool

Éléments à prendre en compte lors de la migration d'un schéma et de code SQL Server vers Amazon RDS pour SQL Server :

- AWS SCT peut convertir l'agent SQL Server pour fournir des plannings, des alertes et des tâches sur une instance de base de données Amazon RDS for SQL Server. Après la conversion, vous pouvez utiliser une instance de base de données Amazon RDS pour SQL Server avec SQL Server Reporting Service (SSRS), SQL Server Analysis Services (SSAS) et SQL Server Integration Services (SSIS).
- Actuellement, Amazon RDS ne prend pas en charge les points de terminaison T-SQL supplémentaires ou SQL Server Service Broker qui nécessitent que vous exécutiez la commande `CREATE ENDPOINT`.
- Amazon RDS offre une prise en charge limitée des serveurs associés. Lors de la conversion du code d'application SQL Server qui utilise des serveurs liés, AWS SCT convertit le code de



l'application. Toutefois, assurez-vous de vérifier le comportement des objets qui utilisent des serveurs liés avant d'exécuter le code converti.

- Always-On est utilisé.
- Le rapport AWS SCT d'évaluation fournit les indicateurs du serveur pour la conversion. Ces metrics sur votre instance SQL Server sont les suivantes :
  - La mise en miroir de données est utilisée.
  - L'envoi de journaux SQL Server est configuré.
  - Un cluster de basculement est utilisé.
  - La messagerie de base de données est configurée.
  - Le service de recherche en texte intégral est utilisé. Amazon RDS pour SQL Server possède une recherche en texte intégral limitée et ne prend pas en charge les recherches sémantiques.
  - Data Quality Service (DQS) est installé. Amazon RDS ne prend pas en charge le DQS. Nous vous recommandons donc d'installer SQL Server sur une instance Amazon EC2.

## Privilèges pour RDS for SQL Server en tant que cible

Pour migrer vers RDS pour SQL Server, créez un utilisateur de base de données, puis accordez les privilèges requis pour chaque base de données. Vous pouvez utiliser l'exemple de code suivant.

```
CREATE LOGIN user_name WITH PASSWORD 'your_password';

USE db_name
CREATE USER user_name FOR LOGIN user_name
GRANT VIEW DEFINITION TO user_name
GRANT VIEW DATABASE STATE TO user_name
GRANT CREATE SCHEMA TO user_name;
GRANT CREATE TABLE TO user_name;
GRANT CREATE VIEW TO user_name;
GRANT CREATE TYPE TO user_name;
GRANT CREATE DEFAULT TO user_name;
GRANT CREATE FUNCTION TO user_name;
GRANT CREATE PROCEDURE TO user_name;
GRANT CREATE ASSEMBLY TO user_name;
GRANT CREATE AGGREGATE TO user_name;
GRANT CREATE FULLTEXT CATALOG TO user_name;
GRANT CREATE SYNONYM TO user_name;
GRANT CREATE XML SCHEMA COLLECTION TO user_name;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez ensuite *db\_name* par le nom de votre base de données cible. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

## Sources d'entrepôts de données pour AWS Schema Conversion Tool

AWS SCT peut convertir les schémas des entrepôts de données sources suivants en une cible prise en charge. Pour plus d'informations sur les autorisations, les connexions et les éléments AWS SCT pouvant être convertis pour une utilisation avec la base de données cible ou l'entrepôt de données, consultez les détails ci-dessous.

### Rubriques

- [Connecter Amazon Redshift à AWS Schema Conversion Tool](#)
- [Connecter Azure Synapse Analytics à AWS Schema Conversion Tool](#)
- [Connexion à Google BigQuery avec AWS Schema Conversion Tool](#)
- [Connecter la base de données Greenplum à AWS Schema Conversion Tool](#)
- [Connexion à Netezza avec AWS Schema Conversion Tool](#)
- [Connecter Oracle Data Warehouse à AWS SCT](#)
- [Connexion à un entrepôt de données Snowflake avec AWS Schema Conversion Tool](#)
- [Connexion à un entrepôt de données SQL Server à l'aide du AWS Schema Conversion Tool](#)
- [Connexion à un entrepôt de données Teradata à l'aide du AWS Schema Conversion Tool](#)
- [Connexion aux bases AWS Schema Conversion Tool de données Vertica](#)

## Connecter Amazon Redshift à AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour optimiser votre cluster Amazon Redshift. AWS SCT vous fournit des recommandations sur la sélection des clés de distribution et de tri pour votre cluster Amazon Redshift. Vous pouvez considérer le projet d'optimisation Amazon Redshift comme un AWS SCT projet dont la source et la cible pointent vers les différents clusters Amazon Redshift.

### Privilèges accordés à Amazon Redshift en tant que base de données source

Les privilèges suivants sont requis pour utiliser Amazon Redshift comme source :

- UTILISATION SUR LE SCHÉMA *<schema\_name>*
- SÉLECTIONNEZ SUR TOUTES LES TABLES DU SCHÉMA *<schema\_name>*
- SELECT ON PG\_CATALOG.PG\_STATISTIC
- SELECT ON SVV\_TABLE\_INFO
- SELECT ON TABLE STV\_BLOCKLIST
- SELECT ON TABLE STV\_TBL\_PERM
- SÉLECTIONNEZ SUR SYS\_SERVERLESS\_USAGE
- SÉLECTIONNEZ SUR PG\_DATABASE\_INFO
- SÉLECTIONNEZ SUR PG\_STATISTIC

Dans les exemples précédents, remplacez l'*<schema\_name>* espace réservé par le nom du schéma source.

Pour connaître les privilèges requis pour Amazon Redshift en tant que cible, consultez. [Autorisations pour Amazon Redshift en tant que cible](#)

## Connexion à Amazon Redshift en tant que source

Suivez la procédure suivante pour vous connecter à votre base de données source Amazon Redshift avec le. AWS Schema Conversion Tool

Pour vous connecter à une base de données source Amazon Redshift

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Amazon Redshift, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.

2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir les informations de connexion à la base de données source Amazon Redshift, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Saisissez le nom de la base de données Amazon Redshift.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Vérifier le certificat du serveur : sélectionnez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li><li>• Trust store : emplacement d'un trust store contenant des certificats. Pour que cet emplacement apparaisse ici, assurez-vous de l'ajouter dans les paramètres généraux.</li></ul> <p>Pour plus d'informations sur la prise en charge du protocole SSL pour Amazon Redshift, consultez <a href="#">Configurer les options de sécurité pour les connexions</a>.</p>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Trajectoire du pilote Redshift	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Paramètres d'optimisation d'Amazon Redshift

Pour modifier les paramètres d'optimisation d'Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Amazon Redshift, puis Amazon Redshift — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour l'optimisation d'Amazon Redshift.

Les paramètres d'optimisation d'Amazon Redshift AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même si le nombre de tables est supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour choisir la stratégie de migration.

AWS recommande d'utiliser différents clusters comme source et cible pour votre projet d'optimisation. Avant de commencer le processus d'optimisation Amazon Redshift, vous devez créer une copie de votre cluster Amazon Redshift source. Vous pouvez inclure vos données sources dans cette copie ou créer un cluster vide.

Pour Stratégie de migration, choisissez Migration vers une copie pour inclure les données de votre cluster source dans le cluster cible.

Pour la stratégie de migration, choisissez Migration vers une table rase pour passer en revue les suggestions d'optimisation. Après avoir accepté ces suggestions, migrez vos données sources vers le cluster cible.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous avez sélectionné l'option Utiliser le codage par compression.

- Pour travailler avec l'optimisation automatique des tables.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour utiliser uniquement l'optimisation automatique des tables, choisissez Stratégies d'optimisation dans le volet de gauche. Sélectionnez ensuite Utiliser le réglage automatique des tables Amazon Redshift, puis sélectionnez Aucun pour la stratégie de sélection des clés initiale.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur asymétrique est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données dont vous souhaitez analyser les statistiques de requête.

## Connecter Azure Synapse Analytics à AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application d'Azure Synapse Analytics vers Amazon Redshift.



## Privilèges pour Azure Synapse Analytics en tant que base de données source

Les privilèges suivants sont requis pour utiliser un entrepôt de données Azure Synapse Analytics comme source :

- VIEW DEFINITION
- VIEW DATABASE STATE

Appliquez les privilèges pour chaque base de données dont vous convertissez le schéma.

## Connexion à Azure Synapse Analytics en tant que source

Utilisez la procédure suivante pour vous connecter à votre entrepôt de données Azure Synapse Analytics avec le AWS Schema Conversion Tool.

Pour vous connecter à un entrepôt de données Azure Synapse Analytics en tant que source

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Azure Synapse Analytics, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion pour l'entrepôt de données Azure Synapse Analytics, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du service DNS (Domain Name Service) ou l'adresse IP du serveur de la base de données source.
pool SQL	Entrez le nom du pool SQL Azure.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT, vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Certificat de serveur de confiance : choisissez cette option pour approuver le certificat de serveur.</li><li>• Trust store : un trust store que vous configurez dans les paramètres globaux.</li></ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Paramètres de conversion d'Azure Synapse Analytics vers Amazon Redshift

Pour modifier les paramètres de conversion d'Azure Synapse Analytics vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Azure Synapse, puis Azure Synapse — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Azure Synapse Analytics vers Amazon Redshift.

Les paramètres de conversion d'Azure Synapse Analytics vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'

AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion d'Azure Synapse Analytics vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion d'Azure Synapse Analytics vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Azure Synapse, puis Azure Synapse — Amazon Redshift. Dans le volet

de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion entre Azure Synapse Analytics et Amazon Redshift.

Les paramètres AWS SCT d'optimisation de la conversion d'Azure Synapse Analytics vers Amazon Redshift incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion à Google BigQuery avec AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application BigQuery vers Amazon Redshift.

### Privilèges pour BigQuery en tant que source

Pour utiliser un entrepôt de BigQuery données comme source dans AWS SCT, créez un compte de service. Dans Google Cloud, les applications utilisent des comptes de service pour effectuer des appels d'API autorisés. Les comptes de service sont différents des comptes utilisateurs. Pour plus d'informations, consultez la section [Comptes de service](#) dans la documentation de Google Cloud Identity and Access Management.

Assurez-vous d'attribuer les rôles suivants à votre compte de service :

- BigQuery Admin

## • Storage Admin

Le BigQuery Admin rôle fournit les autorisations nécessaires pour gérer toutes les ressources du projet. AWS SCT utilise ce rôle pour charger vos BigQuery métadonnées dans le projet de migration.

Le Storage Admin rôle garantit le contrôle total des objets de données et des compartiments. Vous trouverez ce rôle sous Cloud Storage. AWS SCT utilise ce rôle pour extraire vos données d'Amazon Redshift, BigQuery puis les charger dans celui-ci.

Pour créer un fichier clé de compte de service

1. Connectez-vous à la console de gestion Google Cloud à l'adresse <https://console.cloud.google.com/>.
2. Sur la page de [BigQuery l'API](#), sélectionnez Activer. Ignorez cette étape si l'API est activée.
3. Sur la page [Comptes de service](#), choisissez votre projet, puis choisissez Créer un compte de service.
4. Sur la page des détails du compte de service, entrez une valeur descriptive pour le nom du compte de service. Choisissez Créer et continuer. La page Accorder à ce compte de service l'accès au projet s'ouvre.
5. Pour Sélectionner un rôle, choisissez BigQuery, puis choisissez BigQuery Admin.
6. Choisissez Ajouter un autre rôle. Pour Sélectionner un rôle, choisissez Cloud Storage, puis Storage Admin.
7. Choisissez Continuer, puis cliquez sur Terminé.
8. Sur la page [Comptes de service](#), choisissez le compte de service que vous avez créé.
9. Choisissez Clés, puis choisissez Créer une nouvelle clé pour Ajouter une clé.
10. Choisissez JSON, puis Create. Choisissez le dossier dans lequel enregistrer votre clé privée ou sélectionnez le dossier par défaut pour les téléchargements dans votre navigateur.

Pour extraire des données d'un entrepôt de BigQuery données, AWS SCT utilise le dossier bucket de Google Cloud Storage. Créez ce compartiment avant de commencer la migration des données. Entrez le chemin d'accès à votre dossier de bucket Google Cloud Storage dans la boîte de dialogue Créer une tâche locale. Pour de plus amples informations, veuillez consulter [Création, exécution et surveillance d'une AWS SCT tâche](#).

## Connexion à BigQuery en tant que source

Utilisez la procédure suivante pour vous connecter à votre BigQuery projet source avec le AWS Schema Conversion Tool.

Pour se connecter à un entrepôt de données BigQuery source

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez BigQuery, puis cliquez sur Suivant.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de la connexion, entrez le nom de votre BigQuery projet. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Pour Chemin clé, entrez le chemin d'accès au fichier clé du compte de service. Pour plus d'informations sur la création de ce fichier, consultez [Privilèges pour BigQuery en tant que source](#).
5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre BigQuery projet source.
6. Choisissez Connect pour vous connecter à votre BigQuery projet source.

## Restrictions relatives à l'utilisation BigQuery en tant que source pour AWS SCT

Les restrictions suivantes s'appliquent lors de l'utilisation en BigQuery tant que source pour AWS SCT :

- AWS SCT ne prend pas en charge la conversion de sous-requêtes dans les fonctions analytiques.
- Vous ne pouvez pas l' AWS SCT utiliser pour convertir BigQuery SELECT AS STRUCT des SELECT AS VALUE instructions.
- AWS SCT ne prend pas en charge la conversion des types de fonctions suivants :
  - Agrégat approximatif
  - Bit
  - Débogage
  - Requête fédérée
  - Géographie
  - Hachage



- Mathématiques
- Filet
- Agrégat statistique
- UUID
- AWS SCT fournit un support limité pour la conversion des fonctions de chaîne.
- AWS SCT ne prend pas en charge la conversion des UNNEST opérateurs.
- Vous ne pouvez pas convertir les opérations de jointure corrélées en AWS SCT.
- AWS SCT ne prend pas en charge la conversion des OFFSET clauses QUALIFY WINDOWLIMIT,, et.
- Vous ne pouvez pas l'utiliser AWS SCT pour convertir des expressions de table communes récursives.
- AWS SCT ne prend pas en charge la conversion d'INSERT instructions contenant des sous-requêtes dans des VALUES clauses.
- AWS SCT ne prend pas en charge la conversion des UPDATE instructions pour les champs imbriqués et les enregistrements répétés.
- Vous ne pouvez pas l'utiliser AWS SCT pour convertir STRUCT des types de ARRAY données.

## BigQuery vers les paramètres de conversion d'Amazon Redshift

BigQuery Pour modifier les paramètres de conversion d'Amazon Redshift, choisissez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Google BigQuery, puis Google BigQuery — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la BigQuery conversion vers Amazon Redshift.

BigQuery vers les paramètres de conversion Amazon Redshift en AWS SCT incluant des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## BigQuery vers les paramètres d'optimisation de conversion d'Amazon Redshift

BigQuery Pour modifier les paramètres d'optimisation des conversions d'Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Google BigQuery, puis Google BigQuery — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour BigQuery la conversion vers Amazon Redshift.

BigQuery vers les paramètres d'optimisation de conversion d'Amazon Redshift en AWS SCT incluant des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre

minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connecter la base de données Greenplum à AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de la base de données Greenplum vers Amazon Redshift.

### Privilèges pour la base de données Greenplum en tant que source

Les privilèges suivants sont requis pour utiliser la base de données Greenplum en tant que source :

- CONNEXION À LA BASE DE DONNÉES *<database\_name>*
- UTILISATION SUR LE SCHÉMA *<schema\_name>*
- SÉLECTIONNEZ SUR *<schema\_name>.<table\_name>*

- SÉLECTIONNEZ SUR LA SÉQUENCE *<schema\_name>.<sequence\_name>*

Dans l'exemple précédent, remplacez les espaces réservés comme suit :

- Remplacez *database\_name* par le nom de la base de données source.
- Remplacez *schema\_name* par le nom du schéma source.
- Remplacez *table\_name* par le nom de la table source.
- Remplacez *sequence\_name* par le nom de la séquence.

## Connexion à la base de données Greenplum en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Greenplum avec AWS SCT.

Pour vous connecter à une base de données source Greenplum

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez SAP ASE, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations d'identification de la base de données source Greenplum, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Entrez le nom de la base de données Greenplum.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT, vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Vérifier le certificat du serveur : sélectionnez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li><li>• Trust store : emplacement d'un trust store contenant des certificats.</li></ul>

Paramètre	Action
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.
Chemin du pilote de base de données Greenplum	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Paramètres de conversion de Greenplum vers Amazon Redshift

Pour modifier les paramètres de conversion de Greenplum vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Greenplum, puis Greenplum — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion de Greenplum vers Amazon Redshift.

Les paramètres de conversion de Greenplum vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.



- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion de Greenplum vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de Greenplum vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Greenplum, puis Greenplum — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion de Greenplum vers Amazon Redshift.

Les paramètres AWS SCT d'optimisation de la conversion de Greenplum vers Amazon Redshift incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion à Netezza avec AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de Netezza vers Amazon Redshift.

### Privilèges pour Netezza en tant que source

Les privilèges suivants sont requis pour utiliser Netezza en tant que source :

- sélectionnez sur la vue `system.definition_schema.system`
- sélectionnez sur la table `system.definition_schema.system`
- sélectionnez sur la table `system.definition_schema.management`
- liste sur `<database_name>`
- liste sur `<schema_name>`
- liste sur `<database_name> .all.table`
- liste sur la table `<database_name> .all.external`
- liste sur `<database_name> .all.view`
- liste sur la vue `<database_name> .all.materialized`
- liste sur `<database_name> .all.procedure`
- liste sur `<database_name> .all.sequence`
- liste sur `<database_name> .all.function`
- liste sur `<database_name> .all.aggregate`

Dans l'exemple précédent, remplacez les espaces réservés comme suit :

- Remplacez `database_name` par le nom de la base de données source.
- Remplacez `schema_name` par le nom du schéma source.

AWS SCT nécessite l'accès aux tables et vues système suivantes. Vous pouvez accorder l'accès à ces objets au lieu d'accorder l'accès à `system.definition_schema.system view` et `system.definition_schema.system tables` dans la liste précédente.

- sélectionnez sur `system.definition_schema. _t_aggregate`
- sélectionnez sur `system.definition_schema. _t_class`
- sélectionnez sur `system.definition_schema. _t_contrainte`

- sélectionnez sur `system.definition_schema. _t_const_relattr`
- sélectionnez sur `system.definition_schema. _t_base de données`
- sélectionnez sur `system.definition_schema. _t_grobj_priv`
- sélectionnez sur `system.definition_schema. _t_groupes`
- sélectionnez sur `system.definition_schema. _t_hist_config`
- sélectionnez sur `system.definition_schema. _t_objet`
- sélectionnez sur `system.definition_schema. _t_object_classes`
- sélectionnez sur `system.definition_schema. _t_proc`
- sélectionnez sur `system.definition_schema. _t_type`
- sélectionnez sur `system.definition_schema. _t_utilisateur`
- sélectionnez sur `system.definition_schema. _t_usobj_priv`
- sélectionnez sur `system.definition_schema. _vt_sequence`
- sélectionnez sur `system.definition_schema. _v_aggregate`
- sélectionnez sur `system.definition_schema. _v_constraint_depends`
- sélectionnez sur `system.definition_schema. _v_base de données`
- sélectionnez sur `system.definition_schema. _v_datatype`
- sélectionnez sur `system.definition_schema. _v_dslice`
- sélectionnez sur `system.definition_schema. fonction _v`
- sélectionnez sur `system.definition_schema. _v_group`
- sélectionnez sur `system.definition_schema. _v_obj_relation`
- sélectionnez sur `system.definition_schema. _v_obj_relation_xdb`
- sélectionnez sur `system.definition_schema. _v_procédure`
- sélectionnez sur `system.definition_schema. Colonne _v_relation_`
- sélectionnez sur `system.definition_schema. _v_relation_keydata`
- sélectionnez sur `system.definition_schema. classes _v_relobj`
- sélectionnez sur `system.definition_schema. _v_schema_xdb`
- sélectionnez sur `system.definition_schema. _v_séquence`
- sélectionnez sur `system.definition_schema. _v_synonyme`
- sélectionnez sur `system.definition_schema. _v_system_info`
- sélectionnez sur `system.definition_schema. _v_sys_contrainte`

- sélectionnez sur `system.definition_schema._v_sys_object_dslice_info`
- sélectionnez sur `system.definition_schema._v_sys_utilisateur`
- sélectionnez sur `system.definition_schema._v_table`
- sélectionnez sur `system.definition_schema._v_table_contrainte`
- sélectionnez sur `system.definition_schema._v_table_dist_map`
- sélectionnez sur `system.definition_schema._v_table_organiser_colonne`
- sélectionnez sur `system.definition_schema._v_table_storage_stat`
- sélectionnez sur `system.definition_schema._v_utilisateur`
- sélectionnez sur `system.definition_schema._v_view`
- sélectionnez sur `system.information_schema.Colonne_v_relation_`
- sélectionnez sur `system.information_schema._v_table`
- sélectionnez `$hist_column_access_*`

## Connexion à Netezza en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Netezza avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Netezza

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Netezza, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Netezza, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.

Paramètre	Action
Trajectoire du conducteur Netezza	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Configuration de la réplication continue des données

Après avoir converti vos schémas de base de données Netezza et les avoir appliqués à votre base de données Amazon Redshift, vous pouvez migrer les données à l'aide d'agents d'extraction de données. AWS SCT L'agent extrait vos données et les télécharge dans votre compartiment Amazon S3. Vous pouvez ensuite les utiliser AWS SCT pour copier les données d'Amazon S3 vers Amazon Redshift.

Si les données de votre base de données source changent au cours du processus de migration, vous pouvez enregistrer les modifications en cours avec vos agents d'extraction de AWS SCT données. Vous pouvez ensuite répliquer ces modifications en cours dans votre base de données cible après avoir terminé la migration initiale des données. Ce processus est appelé réplication continue des données ou capture des données de modification (CDC).

Pour configurer la réplication continue des données pour les migrations de Netezza vers Amazon Redshift

1. Dans votre base de données source, créez une base de données historique. Vous pouvez utiliser l'exemple de code suivant dans l'interface de ligne de commande (CLI) de Netezza.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

Dans l'exemple précédent, remplacez *history\_database\_name* par le nom de votre base de données d'historique. Remplacez-le ensuite *load\_user* par le nom de l'utilisateur que vous avez défini pour charger les données d'historique dans la base de données. Remplacez ensuite *histdb\_owner* par le nom de l'utilisateur que vous avez défini comme propriétaire de la base de données d'historique. Assurez-vous d'avoir déjà créé cet utilisateur et d'avoir accordé l'`CREATE DATABASE` autorisation. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

2. Configurez la journalisation de l'historique. Pour ce faire, utilisez l'exemple de code suivant.

```
CREATE HISTORY CONFIGURATION history_configuration_name HISTTYPE QUERY
  DATABASE history_database_name USER load_user PASSWORD your_password COLLECT
  PLAN, COLUMN
  LOADINTERVAL 1 LOADMINTHRESHOLD 0 LOADMAXTHRESHOLD 0 STORAGELIMIT 25
  LOADRETRY 2 VERSION 1;
```

Dans l'exemple précédent, remplacez *history\_configuration\_name* et *history\_database\_name* par les noms de votre configuration d'historique et de votre base de données d'historique. Remplacez-le ensuite *load\_user* par le nom de l'utilisateur que vous avez défini pour charger les données d'historique dans la base de données. Remplacez-le ensuite *your\_password* par un mot de passe sécurisé.

3. Accordez des autorisations de lecture pour toutes les tables de la base de données d'historique. Vous pouvez utiliser l'exemple de code suivant pour accorder l'`SELECT` autorisation.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Dans l'exemple précédent, remplacez *history\_database\_name* par le nom de votre base de données d'historique. *your\_user* Remplacez-le ensuite par le nom de l'utilisateur disposant d'autorisations minimales pour utiliser votre base de données Netezza. Vous utilisez les informations d'identification de cet utilisateur de base de données dans AWS SCT.

4. Collectez des statistiques pour chaque table de votre schéma source afin d'obtenir des informations sur la cardinalité des colonnes. Vous pouvez utiliser la commande suivante pour générer des statistiques dans votre base de données d'historique.



```
GENERATE STATISTICS on "schema_name". "table_name";
```

Dans l'exemple précédent, remplacez *schema\_name* et *table\_name* par le nom du schéma et de la table de votre base de données.

5. Vérifiez que vous avez rempli les conditions requises en exécutant la requête suivante :

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Dans l'exemple précédent, remplacez *history\_database\_name* et *history\_schema\_name* par le nom de votre base de données d'historique et de votre schéma. Remplacez ensuite *N* par le numéro de version de votre base de données d'historique. Pour plus d'informations sur les versions des bases de données historiques, consultez la [documentation IBM Netezza](#).

6. Installez vos agents d'extraction de données. Pour de plus amples informations, veuillez consulter [Installation d'agents d'extraction](#).

Assurez-vous que le {working.folder} paramètre du settings.properties fichier pour toutes les instances de l'extracteur pointe vers le même dossier. Dans ce cas, vos extracteurs peuvent coordonner la session CDC et utiliser un point de transaction unique pour toutes les sous-tâches.

7. Enregistrez votre agent d'extraction de données. Pour de plus amples informations, veuillez consulter [Enregistrement des agents d'extraction auprès du AWS Schema Conversion Tool](#).
8. Créez votre tâche CDC. Pour de plus amples informations, veuillez consulter [Création, exécution et surveillance d'une AWS SCT tâche](#).
  - a. Ouvrez votre projet dans AWS SCT. Dans le volet de gauche, choisissez votre table source. Ouvrez le menu contextuel (clic droit), puis choisissez Créer une tâche locale.
  - b. Dans Nom de la tâche, entrez un nom descriptif pour votre tâche de migration de données.
  - c. Pour le mode migration, choisissez Extraire, uploader et copier.
  - d. Sélectionnez Activer le CDC.
  - e. Choisissez l'onglet Paramètres du CDC et définissez la portée et le calendrier des sessions du CDC.
  - f. Choisissez Tâche de test pour vérifier que vous pouvez vous connecter à votre dossier de travail, à votre compartiment Amazon S3 et à votre entrepôt de données Amazon Redshift.

- g. Choisissez Créer pour créer votre tâche.
  - h. Cliquez sur l'onglet Tâches, choisissez votre tâche dans la liste, puis sélectionnez Démarrer.
9. La AWS SCT tâche assure la cohérence transactionnelle dans la base de données cible. L'agent d'extraction de données réplique les transactions depuis la source dans l'ordre des numéros de transaction.

Si vous arrêtez l'une des sessions de migration ou si elle échoue, le traitement CDC s'arrête également.

## Paramètres de conversion de Netezza vers Amazon Redshift

Pour modifier les paramètres de conversion de Netezza vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Netezza, puis Netezza — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion de Netezza vers Amazon Redshift.

Les paramètres de conversion de Netezza vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement

choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'affiche. AWS SCT applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion de Netezza vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de Netezza vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Netezza, puis Netezza — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion de Netezza vers Amazon Redshift.

Les paramètres d'optimisation de la conversion de Netezza vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations,

consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.

- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connecter Oracle Data Warehouse à AWS SCT

Vous pouvez les utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application d'Oracle Data Warehouse vers Amazon Redshift ou Amazon Redshift et les utiliser en combinaison AWS Glue .

### Privilèges pour Oracle Data Warehouse en tant que source

Les privilèges suivants sont requis pour utiliser Oracle Data Warehouse comme source :

- connect
- select\_catalog\_role
- select any dictionary

### Connexion à Oracle Data Warehouse en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source d'entrepôt de données Oracle avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Oracle Data Warehouse

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Oracle, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à l'entrepôt de données source Oracle, suivez les instructions suivantes :

Paramètre	Action
Type	<p>Choisissez le type de connexion à la base de données. Selon le type choisi, fournissez les informations complémentaires suivantes :</p> <ul style="list-style-type: none"> <li>• SID           <ul style="list-style-type: none"> <li>• Nom du serveur : nom du système de noms de domaine (DNS) ou adresse IP de votre serveur de base de données source.</li> <li>• Server port : Port utilisé pour vous connecter au serveur de la base de données source.</li> <li>• Oracle SID : ID du système Oracle (SID). Pour trouver le SID Oracle, soumettez la requête suivante à votre base de données Oracle :               <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> </li> </ul> </li> <li>• Service Name</li> </ul>

Paramètre	Action
	<ul style="list-style-type: none"><li>• Server name : Nom DNS ou adresse IP du serveur de votre base de données source.</li><li>• Server port : Port utilisé pour vous connecter au serveur de la base de données source.</li><li>• Service Name : Nom du service Oracle auquel se connecter.</li><li>• Alias TNS<ul style="list-style-type: none"><li>• TNS file path : Chemin d'accès au fichier qui contient les informations de connexion de nom TNS (Transparent Network Substrate).</li><li>• TNS file path : Alias TNS de ce fichier à utiliser pour vous connecter à la base de données source.</li></ul></li><li>• Identifiant de connexion TNS<ul style="list-style-type: none"><li>• Identifiant de connexion TNS : identifiant des informations de connexion TNS enregistrées.</li></ul></li></ul>
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>

Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Authentification SSL : sélectionnez cette option pour utiliser l'authentification SSL pour la connexion.</li> <li>• Trust store : emplacement d'un trust store contenant des certificats.</li> <li>• Magasin de clés : emplacement d'un magasin de clés contenant une clé privée et des certificats. Cette valeur est obligatoire si l'authentification SSL est sélectionnée ; sinon, elle est facultative.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Chemin du pilote Oracle	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.



## Paramètres de conversion d'Oracle Data Warehouse vers Amazon Redshift

Pour modifier les paramètres de conversion d'Oracle Data Warehouse vers Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Oracle, puis Oracle — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion d'Oracle Data Warehouse vers Amazon Redshift.

Les paramètres de conversion d'Oracle Data Warehouse vers Amazon Redshift AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.

- Pour convertir les fonctions de formatage des types de données telles que TO\_CHAR TO\_DATE, et TO\_NUMBER avec des éléments de format datetime non pris en charge par Amazon Redshift. Par défaut, AWS SCT utilise les fonctions du pack d'extension pour émuler l'utilisation de ces éléments de format non pris en charge dans le code converti.

Le modèle de format datetime d'Oracle inclut davantage d'éléments que les chaînes de format datetime d'Amazon Redshift. Lorsque votre code source inclut uniquement des éléments de format datetime pris en charge par Amazon Redshift, vous n'avez pas besoin des fonctions du pack d'extension dans le code converti. Pour éviter d'utiliser les fonctions du pack d'extension dans le code converti, sélectionnez les éléments de format Datetype que vous utilisez dans le code Oracle similaires aux chaînes de format datetime d'Amazon Redshift. Dans ce cas, le code converti fonctionne plus rapidement.

Le modèle de format numérique d'Oracle inclut davantage d'éléments que les chaînes de format numérique d'Amazon Redshift. Lorsque votre code source inclut uniquement des éléments de format numérique pris en charge par Amazon Redshift, vous n'avez pas besoin des fonctions du pack d'extension dans le code converti. Pour éviter d'utiliser les fonctions du pack d'extension dans le code converti, sélectionnez Les éléments de format numérique que vous utilisez dans le code Oracle sont similaires aux chaînes de format numérique dans Amazon Redshift. Dans ce cas, le code converti fonctionne plus rapidement.

- Pour convertir des fonctions Oracle LEAD et LAG analytiques. Par défaut, AWS SCT déclenche une action pour chaque LAG fonction LEAD et.

Lorsque votre code source n'utilise pas les valeurs par défaut pour le décalage dans ces fonctions, AWS SCT vous pouvez émuler l'utilisation de ces fonctions avec la NVL fonction. Pour ce faire, sélectionnez Utiliser la fonction NVL pour émuler le comportement des fonctions Oracle LEAD et LAG.

- Pour émuler le comportement des clés primaires et uniques dans votre cluster Amazon Redshift, sélectionnez Émuler le comportement des clés primaires et uniques.

Amazon Redshift n'applique pas les clés uniques et primaires et les utilise uniquement à des fins d'information. Si vous utilisez ces contraintes dans votre code, assurez-vous qu'elles AWS SCT imitent leur comportement dans le code converti.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion d'Oracle Data Warehouse vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion d'Oracle Data Warehouse vers Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Oracle, puis Oracle — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion d'Oracle Data Warehouse vers Amazon Redshift.

Les paramètres d'optimisation de la conversion d'Oracle Data Warehouse vers Amazon Redshift AWS SCT incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique

d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.

- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion à un entrepôt de données Snowflake avec AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de Snowflake vers Amazon Redshift.

### Privilèges pour Snowflake en tant que base de données source

Vous pouvez créer un rôle doté de privilèges et attribuer à ce rôle le nom d'un utilisateur en utilisant le SECURITYADMIN rôle et le contexte de SECURITYADMIN session.

L'exemple suivant crée des privilèges minimaux et les accorde à l'`min_privsutilisateur`.

```
create role role_name;  
grant role role_name to role sysadmin;  
grant usage on database db_name to role role_name;  
grant usage on schema db_name.schema_name to role role_name;  
grant usage on warehouse datawarehouse_name to role role_name;  
grant monitor on database db_name to role role_name;  
grant monitor on warehouse datawarehouse_name to role role_name;  
grant select on all tables in schema db_name.schema_name to role role_name;  
grant select on future tables in schema db_name.schema_name to role role_name;  
grant select on all views in schema db_name.schema_name to role role_name;  
grant select on future views in schema db_name.schema_name to role role_name;  
grant select on all external tables in schema db_name.schema_name to role role_name;  
grant select on future external tables in schema db_name.schema_name to role role_name;
```

```
grant usage on all sequences in schema db_name.schema_name to role role_name;  
grant usage on future sequences in schema db_name.schema_name to role role_name;  
grant usage on all functions in schema db_name.schema_name to role role_name;  
grant usage on future functions in schema db_name.schema_name to role role_name;  
grant usage on all procedures in schema db_name.schema_name to role role_name;  
grant usage on future procedures in schema db_name.schema_name to role role_name;  
create user min_privs password='real_user_password'  
DEFAULT_ROLE = role_name DEFAULT_WAREHOUSE = 'datawarehouse_name';  
grant role role_name to user min_privs;
```

Dans l'exemple précédent, remplacez les espaces réservés comme suit :

- *role\_name* Remplacez-le par le nom d'un rôle doté de privilèges de lecture seule.
- Remplacez *db\_name* par le nom de la base de données source.
- Remplacez *schema\_name* par le nom du schéma source.
- Remplacez *datawarehouse\_name* par le nom d'un entrepôt de données requis.
- Remplacez *min\_privs* par le nom d'un utilisateur disposant de privilèges minimaux.

Les DEFAULT\_WAREHOUSE paramètres DEFAULT\_ROLE et sont sensibles aux touches.

## Configuration de l'accès sécurisé à Amazon S3

Les politiques de sécurité et de gestion des accès pour un compartiment Amazon S3 permettent à Snowflake d'accéder au compartiment S3, de lire des données et d'y écrire des données. Vous pouvez configurer un accès sécurisé à un compartiment Amazon S3 privé à l'aide du type d'OBJECT INTEGRATION objet Snowflake. Un objet d'intégration de stockage Snowflake délègue la responsabilité de l'authentification à une entité de gestion des identités et des accès Snowflake.

Pour plus d'informations, consultez [la section Configuration d'une intégration de stockage Snowflake pour accéder à Amazon S3](#) dans la documentation de Snowflake.

## Connexion à Snowflake en tant que source

Suivez la procédure ci-dessous pour vous connecter à votre base de données source à l'aide du AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Snowflake

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.

## 2. Choisissez Snowflake, puis Next.

La boîte de dialogue Ajouter une source apparaît.

## 3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.

## 4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :

- Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :

1. Pour AWS Secret, choisissez le nom du secret.

2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à l'entrepôt de données source Snowflake, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Entrez le nom de la base de données Snowflake.
User name et Password	Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.  AWS SCT enregistre votre mot de passe dans un format crypté uniquement si vous le demandez explicitement.

Paramètre	Action
Use SSL	<p>Choisissez cette option si vous souhaitez utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"><li>• Chemin de clé privée : emplacement d'une clé privée.</li><li>• Phrase secrète : phrase secrète de la clé privée.</li></ul> <p>Pour plus d'informations sur le support SSL pour Snowflake , consultez <a href="#">Configurer les options de sécurité pour les connexions</a>.</p>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. Si vous définissez cette option, vous pouvez enregistrer le mot de passe de la base de données. Cela signifie que vous pouvez vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Trajectoire du pilote Snowflake	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.



## Limitations de Snowflake en tant que source

Les limites suivantes s'appliquent à l'utilisation de Snowflake comme source pour : AWS SCT

- Les identificateurs d'objet doivent être uniques dans le contexte du type d'objet et de l'objet parent :  
Base de données

Les identificateurs de schéma doivent être uniques au sein d'une base de données.

### Schémas

Les identifiants d'objets tels que les tables et les vues doivent être uniques au sein d'un schéma.

### Tables/Vues

Les identificateurs de colonne doivent être uniques dans une table.

- Le nombre maximal de tables pour les types de nœud de cluster large et xlarge est de 9 900. Pour les types de nœuds de cluster de 8 x large, le nombre maximum de tables est de 100 000. La limite inclut les tables temporaires, définies par l'utilisateur et créées par Amazon Redshift lors du traitement des requêtes ou de la maintenance du système. Pour plus d'informations, consultez la rubrique [Quotas Amazon Redshift](#) dans le Guide de la gestion du cluster Amazon Redshift.
- Pour les procédures stockées, le nombre maximum d'arguments d'entrée et de sortie est de 32.

## Types de données source pour Snowflake

Vous trouverez ci-dessous les types de données source Snowflake pris en charge lors de l'utilisation AWS SCT et le mappage par défaut vers une cible Amazon Redshift.

Types de données Snowflake	Types de données Amazon Redshift
NOMBRE	NUMÉRIQUE (38)
NUMÉRO (p)	Si p est $\leq 4$ , alors SMALLINT Si p est $\geq 5$ et $\leq 9$ , alors INTEGER Si p est $\geq 10$ et $\leq 18$ , alors BIGINT Si p est $\geq 19$ alors NUMERIC (p)

Types de données Snowflake	Types de données Amazon Redshift
NUMÉRO (p, 0)	Si p est =< 4, alors SMALLINT  Si p est => 5 et =< 9, alors INTEGER  Si p est => 10 et =< 18, alors BIGINT  Si p est => 19 alors : NUMERIC (p,0)
NUMÉRO (p, s)	Si p est => 1 et =< 38, et si s est => 1 et =< 37, alors  NUMERIC(p,s)
FLOAT	FLOAT
TEXT  Caractères Unicode jusqu'à 16 777 216 octets ; jusqu'à 4 octets par caractère.	VARCHAR(MAX)
TEXTE (p)  Caractères Unicode jusqu'à 65 535 octets ; jusqu'à 4 octets par caractère.	Si p est =< 65 535 alors, VARCHAR (p)
TEXTE (p)  Caractères Unicode jusqu'à 16 777 216 octets ; jusqu'à 4 octets par caractère.	Si p est => 65 535 et =< 16 777 216 alors, VARCHAR (MAX)
BINAIRE  Caractères à un octet jusqu'à 8 388 608 octets ; 1 octet par caractère.	VARCHAR(MAX)
BINAIRE (p)  Caractères à un octet jusqu'à 65 535 octets ; 1 octet par caractère.	VARCHAR (p)

Types de données Snowflake	Types de données Amazon Redshift
BINAIRE (p)  Caractères à un octet jusqu'à 8 388 608 octets ; 1 octet par caractère.	VARCHAR(MAX)
BOOLEAN	BOOLEAN
DATE	DATE
TIME  Valeurs temporelles comprises entre 00:00:00 et 23:59:59.999 999999.	VARCHAR(18)
HEURE (f)  Valeurs temporelles comprises entre 00:00:00 et 23:59:59.9 (f).	VARCHAR (n) — 9 + dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

## Paramètres de conversion de Snowflake vers Amazon Redshift

Pour modifier les paramètres de conversion de Snowflake vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Snowflake, puis Snowflake — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion de Snowflake vers Amazon Redshift.

Les paramètres de conversion de Snowflake vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables que AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'affiche. AWS SCT applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion de Snowflake vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de Snowflake vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Snowflake, puis Snowflake — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion de Snowflake vers Amazon Redshift.

Les paramètres d'optimisation de la conversion de Snowflake vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre

minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion à un entrepôt de données SQL Server à l'aide du AWS Schema Conversion Tool

Vous pouvez les utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de Microsoft SQL Server DW vers Amazon Redshift ou Amazon Redshift et les utiliser en combinaison. AWS Glue

### Privilèges pour Microsoft SQL Server Data Warehouse en tant que source

Les privilèges suivants sont requis pour utiliser l'entrepôt de données Microsoft SQL Server comme source :

- VIEW DEFINITION
- VIEW DATABASE STATE
- SÉLECTIONNEZ SUR LE SCHÉMA : *<schema\_name>*

Dans l'exemple précédent, remplacez l'*<source\_schema>* espace réservé par le nom de la source `source_schema`.

Répétez l'octroi pour chaque base de données dont vous convertissez le schéma.

En outre, accordez les privilèges suivants et exécutez l'octroi sur la base de données principale :

- VIEW SERVER STATE

## Limitations de l'entrepôt de données SQL Server en tant que source

L'utilisation de Microsoft SQL Server Parallel Data Warehouse (PDW) comme source n'est actuellement pas prise en charge.

## Connexion à l'entrepôt de données SQL Server en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source SQL Server Data Warehouse avec le AWS Schema Conversion Tool.

Pour vous connecter à une base de données source SQL Server Data Warehouse

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Microsoft SQL Server, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.

2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à l'entrepôt de données source Microsoft SQL Server, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du service DNS (Domain Name Service) ou l'adresse IP du serveur de la base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Instance name	Entrez le nom de l'instance pour l'entrepôt de données SQL Server.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>



Paramètre	Action
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Certificat de serveur de confiance : sélectionnez cette option pour approuver le certificat de serveur.</li> <li>• Trust store : un trust store que vous configurez dans les paramètres globaux.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Chemin du pilote SQL Server	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Paramètres de conversion de SQL Server Data Warehouse vers Amazon Redshift

Pour modifier les paramètres de conversion de SQL Server Data Warehouse vers Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis Paramètres de conversion. Dans la liste supérieure, choisissez Microsoft SQL Server, puis Microsoft SQL Server — Amazon Redshift. AWS SCT affiche

tous les paramètres disponibles pour la conversion de SQL Server Data Warehouse vers Amazon Redshift.

Les paramètres de conversion de SQL Server Data Warehouse vers Amazon Redshift AWS SCT incluent des options pour les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque

partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion de SQL Server Data Warehouse vers Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de SQL Server Data Warehouse vers Amazon Redshift, sélectionnez Paramètres dans AWS SCT, puis Paramètres de conversion. Dans la liste supérieure, choisissez Microsoft SQL Server, puis Microsoft SQL Server — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation de la conversion pour la conversion de SQL Server Data Warehouse vers Amazon Redshift.

Les paramètres d'optimisation de la conversion de SQL Server Data Warehouse vers Amazon Redshift AWS SCT incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique

d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.

- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion à un entrepôt de données Teradata à l'aide du AWS Schema Conversion Tool

Vous pouvez les utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de Teradata vers Amazon Redshift ou Amazon Redshift et les utiliser en combinaison. AWS Glue

### Privilèges accordés à Teradata en tant que source

Les privilèges suivants sont requis pour utiliser Teradata en tant que source :

- SELECT ON DBC
- SÉLECTIONNEZ SUR SYSUDTLIB
- SÉLECTIONNER SUR SYSLIB
- SÉLECTIONNEZ SUR *<source\_database>*
- CRÉER UNE PROCÉDURE SUR *<source\_database>*

Dans l'exemple précédent, remplacez l'*<source\_database>* espace réservé par le nom de la base de données source.

AWS SCT nécessite le privilège CREATE PROCEDURE pour exécuter HELP PROCEDURE sur toutes les procédures de la base de données source. AWS SCT n'utilise pas ce privilège pour créer de nouveaux objets dans votre base de données Teradata source.

## Connexion à Teradata en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Teradata avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Teradata

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Teradata, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Teradata, suivez les instructions suivantes :

Paramètre	Action
Connection name (Nom de la connexion)	Entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.

Paramètre	Action
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Tapez le nom de la base de données Teradata.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT, vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Enregistrer le mot de passe	AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.
Chiffrer les données	Choisissez cette option pour chiffrer les données que vous échangez avec la base de données. Si vous choisissez cette option, le numéro de port 443 est utilisé pour transférer des données cryptées entre AWS SCT et votre base de données Teradata.

Paramètre	Action
Chemin du pilote Teradata	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.
6. Choisissez Connect pour vous connecter à votre base de données source.

## Utilisation de l'authentification LDAP avec une source Teradata

Pour configurer l'authentification LDAP (Lightweight Directory Access Protocol) pour les utilisateurs Teradata qui exécutent Microsoft Active Directory sous Windows, utilisez la procédure suivante.

Dans la procédure suivante, le domaine Active Directory est `test.local.com`. Le serveur Windows l'estDC, et il est configuré avec les paramètres par défaut. Le script suivant crée le compte `test_ldap` Active Directory, et ce compte utilise le `test_ldap` mot de passe.

Pour configurer l'authentification LDAP pour les utilisateurs de Teradata qui exécutent Microsoft Active Directory sous Windows

1. Dans le répertoire `/opt/teradata/tdat/tdgss/site`, modifiez le fichier `TdgssUserConfigFile.xml`. Modifiez la section LDAP comme suit.

```
AuthorizationSupported="no"

LdapServerName="DC.test.local.com"
LdapServerPort="389"
LdapServerRealm="test.local.com"
LdapSystemFQDN="dc= test, dc= local, dc=com"
LdapBaseFQDN="dc=test, dc=local, dc=com"
```



2. Appliquez les modifications en exécutant la configuration comme suit.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Testez la configuration en exécutant la commande suivante.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

La sortie doit ressembler à ce qui suit.

```
LdapGroupBaseFQDN: dc=Test, dc=local, dc=com
LdapUserBaseFQDN: dc=Test, dc=local, dc=com
LdapSystemFQDN: dc= test, dc= local, dc=com
LdapServerName: DC.test.local.com
LdapServerPort: 389
LdapServerRealm: test.local.com
LdapClientUseTls: no
LdapClientTlsReqCert: never
LdapClientMechanism: SASL/DIGEST-MD5
LdapServiceBindRequired: no
LdapClientTlsCRLCheck: none
LdapAllowUnsafeServerConnect: yes
UseLdapConfig: no
AuthorizationSupported: no
FQDN: CN=test, CN=Users, DC=Anthem, DC=local, DC=com
AuthUser: ldap://DC.test.local.com:389/CN=test1,CN=Users,DC=test,DC=local,DC=com
DatabaseName: test
Service: tdsbind
```

4. Redémarrez TPA à l'aide de la commande suivante.

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. Créez le même utilisateur dans la base de données Teradata que dans Active Directory, comme indiqué ci-dessous.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Si vous modifiez le mot de passe utilisateur dans Active Directory pour votre utilisateur LDAP, spécifiez ce nouveau mot de passe lors de la connexion à Teradata en mode LDAP. En mode DEFAULT, vous vous connectez à Teradata en utilisant le nom d'utilisateur LDAP et n'importe quel mot de passe.

## Configuration de la collecte de statistiques dans votre entrepôt de données Teradata source

Pour convertir votre entrepôt de données Teradata source, utilisez AWS SCT des statistiques pour optimiser votre entrepôt de données Amazon Redshift converti. Vous pouvez collecter des statistiques dans le fichier de statistiques AWS SCT ou le télécharger. Pour de plus amples informations, veuillez consulter [Collecte ou téléchargement de statistiques](#).

Pour être sûr de AWS SCT pouvoir collecter des statistiques à partir de votre entrepôt de données, effectuez les tâches préalables suivantes.

Pour collecter des statistiques à partir de votre entrepôt de données Teradata

1. Exécutez la requête suivante pour recueillir les statistiques de toutes les tables de votre entrepôt de données.

```
collect summary statistics on table_name;
```

Dans l'exemple précédent, remplacez *table\_name* par le nom de votre table source. Répétez la requête pour chaque table que vous convertissez.

2. Exécutez la requête suivante pour déterminer la chaîne de compte de l'utilisateur, que vous utiliserez pour convertir votre entrepôt de données.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Activez la journalisation des requêtes pour un utilisateur spécifique à l'aide de la chaîne de compte de l'exemple précédent.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('M$BUSI$S$D$H');
```

Vous pouvez également activer la journalisation des requêtes pour tous les utilisateurs de la base de données.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Une fois que vous avez terminé de collecter les statistiques de l'entrepôt de données, désactivez l'enregistrement des requêtes. Pour ce faire, vous pouvez utiliser l'exemple de code suivant.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

## Collecte de statistiques en mode hors ligne à partir de votre entrepôt de données Teradata source

Après avoir configuré la collecte de statistiques dans votre entrepôt de données Teradata, vous pouvez collecter des statistiques dans votre AWS SCT projet. Vous pouvez également utiliser des scripts BTEQ (Basic Teradata Query) pour collecter des statistiques en mode hors ligne. Vous pouvez ensuite télécharger les fichiers contenant les statistiques collectées dans votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Collecte ou téléchargement de statistiques](#).

Pour collecter des statistiques à partir de votre entrepôt de données Teradata en mode hors ligne

1. Créez le `off-line_stats.bteq` script avec le contenu suivant.

```
.OS IF EXIST column-stats-tera.csv del /F column-stats-tera.csv
.OS IF EXIST table-stats-tera.csv del /F table-stats-tera.csv
.OS IF EXIST column-skew-script-tera.csv del /F column-skew-script-tera.csv
.OS IF EXIST column-skew-stats-tera.csv del /F column-skew-stats-tera.csv
.OS IF EXIST query-stats-tera.csv del /F query-stats-tera.csv
.LOGON your_teradata_server/your_login, your_password
.EXPORT REPORT FILE = table-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
  ''' || OREPLACE(COALESCE(c.DatabaseName, ''), '', '""') || ';' ||
  ''' || OREPLACE(COALESCE(c.TableName, ''), '', '""') || ';' ||
  ''' || TRIM(COALESCE(s.reference_count, '0')) || ';' ||
  ''' || TRIM(COALESCE(CAST(p.RowCount AS BIGINT), '0')) || ';' ||
  ''' || CAST(CAST(w.size_in_mb AS DECIMAL (38,1) FORMAT 'Z9.9') AS VARCHAR(38))
  || ';' ||
  ''' || TRIM(COALESCE(r.stat_fk_dep_count, '0')) || ';' ||
```

```

    '' || CAST(CAST(current_timestamp(0) as timestamp(0) format 'YYYY-MM-
DDBHH:MI:SS') as VARCHAR(19)) || ''
(TITLE
  "database_name";"table_name";"reference_count";"row_count";"size_in_mb";"stat_fk_dep_count"
FROM (select databasename, tablename
      from DBC.tablesv
      where tablekind IN ('T','O')
      and databasename = 'your_database_name'
      ) c
left join
  (select DatabaseName, TableName, max(RowCount) RowCount
   from dbc.tableStatsv
   group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
  (SELECT r.ChildDB as DatabaseName,
   r.ChildTable as TableName,
   COUNT(DISTINCT r.ParentTable) reference_count
   FROM DBC.All_RI_ChildrenV r
   GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
  (SELECT r.ParentDB as DatabaseName,
   r.ParentTable as TableName,
   COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
   FROM DBC.All_RI_ParentsV r
   GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
  (select databasename, tablename,
   sum(currentperm)/1024/1024 as size_in_mb
   from dbc.TableSizeV
   group by 1,2) w
on w.databasename = c.databasename
and w.tablename = c.tablename
WHERE COALESCE(r.stat_fk_dep_count,0) + COALESCE(CAST(p.RowCount AS BIGINT),0) +
  COALESCE(s.reference_count,0) > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-stats-tera.csv

```

```

.SET TITLEDASHES OFF
.SET WIDTH 10000
    ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) || ';' ||

SELECT
    ''' || OREPLACE(COALESCE(trim(tv.DatabaseName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.TableName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.columnname), ''), '', '""') || ';' ||
    ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) ||

';' ||

    ''' || CAST(current_timestamp AS VARCHAR(19)) || ''' (TITLE
    "database_name";"table_name";"column_name";"cardinality";"current_ts")
FROM dbc.columnsv tv
LEFT JOIN
(
    SELECT
        c.DatabaseName AS DATABASE_NAME,
        c.TABLENAME AS TABLE_NAME,
        c.ColumnName AS COLUMN_NAME,
        c.UniqueValueCount AS CARD
    FROM dbc.tablestatsv c
    WHERE c.DatabaseName = 'your_database_name'
    AND c.RowCount <> 0
) t2
ON tv.DATABASENAME = t2.DATABASE_NAME
AND tv.TABLENAME = t2.TABLE_NAME
AND tv.COLUMNNAME = t2.COLUMN_NAME
WHERE t2.card > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-script-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
'SELECT CAST(''''' || TRIM(c.DatabaseName) || ';' || TRIM(c.TABLENAME) || ';' ||
    || TRIM(c.COLUMNNAME) || ';' ||
TRIM(CAST(COALESCE(MAX(cnt) * 1.0 / SUM(cnt), 0) AS NUMBER FORMAT '9.9999')) ||
    || ';' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || '""' AS VARCHAR(512))
AS """"DATABASE_NAME""";""TABLE_NAME"";""COLUMN_NAME"";""SKEWED"";""CURRENT_TS""
FROM(

```

```

SELECT COUNT(*) AS cnt
FROM '' || c.DATABASENAME || '.' || c.TABLENAME ||
'' GROUP BY '' || c.COLUMNNAME || '' t' ||
CASE WHEN ROW_NUMBER() OVER(PARTITION BY c.DATABASENAME
ORDER BY c.TABLENAME DESC, c.COLUMNNAME DESC) <> 1
THEN ' UNION ALL'
ELSE ';' END (TITLE '--SKEWED--')
FROM dbc.columnsv c
INNER JOIN
(SELECT databasename, TABLENAME
FROM dbc.tablesv WHERE tablekind = 'T'
AND databasename = 'your_database_name') t
ON t.databasename = c.databasename
AND t.TABLENAME = c.TABLENAME
INNER JOIN
(SELECT databasename, TABLENAME, columnname FROM dbc.indices GROUP BY 1,2,3
WHERE TRANSLATE_CHK (databasename USING LATIN_TO_UNICODE) + TRANSLATE_CHK
(TABLENAME USING LATIN_TO_UNICODE) + TRANSLATE_CHK (columnname USING
LATIN_TO_UNICODE) = 0
) i
ON i.databasename = c.databasename
AND i.TABLENAME = c.TABLENAME
AND i.columnname = c.columnname
WHERE c.ColumnType NOT IN ('CO', 'JN', 'N', '++', 'VA', 'UT', 'AN', 'XM', 'A1', 'B0')
ORDER BY c.TABLENAME, c.COLUMNNAME;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

.RUN FILE = column-skew-script-tera.csv

.EXPORT RESET

.EXPORT REPORT FILE = query-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 32000

SELECT
'' || RTRIM(CAST(SqlTextInfo AS VARCHAR(31900))), ';' || ';' ||
TRIM(QueryCount) || ';' ||
TRIM(QueryId) || ';' ||

```

```

TRIM(SqlRowNo) || ';' ||
TRIM(QueryParts) || ';' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || ''
(TITLE
'"query_text";"query_count";"query_id";"sql_row_no";"query_parts";"current_ts"')
FROM
(
  SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
  SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
  ROWS UNBOUNDED PRECEDING) AS topN
  FROM
  (SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
  CASE WHEN
  ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
  SqlRowNo) = 1 AND SqlRowNo = 1
  THEN 1 ELSE 0 END AS QueryFirstRow
  FROM (
  SELECT q.QueryId, q.SqlTextInfo, q.SqlRowNo,
  MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
  COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
  FROM DBC.dbqsqltbl q
  INNER JOIN
  (
  SELECT QueryId
  FROM DBC.DBQLogTbl t
  WHERE TRIM(t.StatementType) IN ('SELECT')
  AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
  AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
  TRIM(t.USERNAME) END) IN ('All users') --user_name list
  AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
  GROUP BY 1
  ) t
  ON q.QueryId = t.QueryId
  INNER JOIN
  (
  SELECT QueryId
  FROM DBC.QryLogObjectsV
  WHERE ObjectDatabaseName = 'your_database_name'
  AND ObjectType = 'Tab'
  AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
  GROUP BY 1
  ) r
  ON r.QueryId = t.QueryId
  WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY

```

```

    ) t
  ) t
WHERE SqlTextInfo NOT LIKE '%"%"'
) q
WHERE
QueryParts >=1
AND topN <= 50
ORDER BY QueryCount DESC, QueryId, SqlRowNo
QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

.EXPORT RESET

.LOGOFF

.QUIT

```

2. Créez le `td_run_bteq.bat` fichier qui exécute le script BTEQ que vous avez créé à l'étape précédente. Utilisez le contenu suivant pour ce fichier.

```

@echo off > off-line_stats1.bteq & setLocal enableDELAYedexpansion
@echo off > off-line_stats2.bteq & setLocal enableDELAYedexpansion

set old1=your_teradata_server
set new1=%1
set old2=your_login
set new2=%2
set old3=your_database_name
set new3=%3
set old4=your_password
set /p new4=Input %2 pass?

for /f "tokens=* delims= " %%a in (off-line_stats.bteq) do (
set str1=%%a
set str1=!str1:%old1%=%new1%!
>> off-line_stats1.bteq echo !str1!
)

for /f "tokens=* delims= " %%a in (off-line_stats1.bteq) do (
set str2=%%a
set str2=!str2:%old2%=%new2%!
>> off-line_stats2.bteq echo !str2!
)

```



```
type nul > off-line_stats1.bteq

for /f "tokens=* delims= " %%a in (off-line_stats2.bteq) do (
set str3=%%a
set str3=!str3:%old3%=!new3%!
>> off-line_stats1.bteq echo !str3!
)

type nul > off-line_stats2.bteq

for /f "tokens=* delims= " %%a in (off-line_stats1.bteq) do (
set str4=%%a
set str4=!str4:%old4%=!new4%!
>> off-line_stats2.bteq echo !str4!
)

del .\off-line_stats1.bteq

echo export starting...

bteq -c UTF8 < off-line_stats.bteq > metadata_export.log

pause
```

3. Créez le `runme.bat` fichier qui exécute le fichier batch que vous avez créé à l'étape précédente. Utilisez le contenu suivant pour ce fichier.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

Dans le `runme.bat` fichier, remplacez *ServerName**UserName*, et *DatabaseName* par les valeurs applicables.

Exécutez ensuite le `runme.bat` fichier. Répétez cette étape pour chaque entrepôt de données que vous convertissez en Amazon Redshift.

Après avoir exécuté ce script, vous recevez trois fichiers contenant des statistiques pour chaque base de données. Vous pouvez télécharger ces fichiers dans votre AWS SCT projet. Pour ce faire, choisissez votre entrepôt de données dans le panneau de gauche de votre projet et ouvrez le menu contextuel (clic droit). Choisissez Upload Statistics.

## Paramètres de conversion de Teradata vers Amazon Redshift

Pour modifier les paramètres de conversion de Teradata vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Teradata, puis Teradata — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion de Teradata vers Amazon Redshift.

Les paramètres de conversion de Teradata vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables qui AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'AWS SCT affiche. AWS SCT Applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

- Pour utiliser une liste explicite de colonnes dans le code converti pour les SELECT \* instructions, sélectionnez Utiliser une déclaration de colonne explicite.
- Pour émuler le comportement des clés primaires et uniques dans votre cluster Amazon Redshift, sélectionnez Émuler le comportement des clés primaires et uniques.

Amazon Redshift n'applique pas les clés uniques et primaires et ne les utilise qu'à des fins d'information. Si vous utilisez ces contraintes dans votre code, assurez-vous qu'elles AWS SCT imitent leur comportement dans le code converti.

- Pour garantir l'unicité des données dans les tables Amazon Redshift cibles. Pour ce faire, sélectionnez Émuler le comportement des tables SET.

Teradata crée des tables en utilisant l'élément de SET syntaxe comme option par défaut. Vous ne pouvez pas ajouter de lignes dupliquées dans un SET tableau. Si votre code source n'utilise pas cette contrainte d'unicité, désactivez cette option. Dans ce cas, le code converti fonctionne plus rapidement.

Si votre code source utilise l'SEToption dans les tables comme contrainte d'unicité, activez cette option. Dans ce cas, AWS SCT réécrit INSERT . . SELECT les instructions dans le code converti pour imiter le comportement de votre base de données source.

## Paramètres d'optimisation de la conversion entre Teradata et Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de Teradata vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Teradata, puis Teradata — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation de conversion pour la conversion de Teradata vers Amazon Redshift.

Les paramètres AWS SCT d'optimisation de la conversion de Teradata vers Amazon Redshift incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques

- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.
- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

## Connexion aux bases AWS Schema Conversion Tool de données Vertica

Vous pouvez l'utiliser AWS SCT pour convertir des schémas, des objets de code et du code d'application de Vertica vers Amazon Redshift.

### Privilèges pour Vertica en tant que source

Les privilèges suivants sont requis pour utiliser Vertica en tant que source :

- UTILISATION SUR LE SCHÉMA *<schema\_name>*
- USAGE ON SCHEMA PUBLIC
- SÉLECTIONNEZ SUR TOUTES LES TABLES DU SCHÉMA *<schema\_name>*
- SÉLECTIONNEZ TOUTES LES SÉQUENCES DU SCHÉMA *<schema\_name>*
- EXÉCUTER SUR TOUTES LES FONCTIONS DU SCHÉMA *<schema\_name>*
- EXÉCUTER SUR LA PROCÉDURE  
*<schema\_name.procedure\_name(procedure\_signature)>*

Dans l'exemple précédent, remplacez les espaces réservés comme suit :

- Remplacez *schema\_name* par le nom du schéma source.
- Remplacez *procedure\_name* par le nom d'une procédure source. Répétez l'autorisation pour chaque procédure que vous convertissez.
- Remplacez *procedure\_signature* par la liste des types d'arguments de procédure séparés par des virgules.

### Connexion à Vertica en tant que source

Utilisez la procédure suivante pour vous connecter à votre base de données source Vertica avec AWS Schema Conversion Tool.

Pour vous connecter à une base de données source Vertica

1. Dans le AWS Schema Conversion Tool, choisissez Ajouter une source.
2. Choisissez Vertica, puis Next.

La boîte de dialogue Ajouter une source apparaît.

3. Dans Nom de connexion, entrez le nom de votre base de données. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Utilisez les informations d'identification de la base de données AWS Secrets Manager ou saisissez-les manuellement :
  - Pour utiliser les informations d'identification de base de données issues de Secrets Manager, suivez les instructions suivantes :
    1. Pour AWS Secret, choisissez le nom du secret.
    2. Choisissez Populer pour renseigner automatiquement toutes les valeurs dans la boîte de dialogue de connexion à la base de données depuis Secrets Manager.

Pour plus d'informations sur l'utilisation des informations d'identification de base de données depuis Secrets Manager, consultez [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

- Pour saisir manuellement les informations de connexion à la base de données source Vertica, suivez les instructions suivantes :

Paramètre	Action
Server name	Entrez le nom du système de noms de domaine (DNS) ou l'adresse IP de votre serveur de base de données source.
Server port	Indiquez le port utilisé pour vous connecter au serveur de la base de données source.
Database (Base de données)	Tapez le nom de la base de données Vertica.
User name et Password	<p>Entrez les informations d'identification de la base de données pour vous connecter à votre serveur de base de données source.</p> <p>AWS SCT utilise le mot de passe pour se connecter à votre base de données source uniquement lorsque vous choisissez de vous connecter à votre base de données dans le cadre d'un projet. Pour éviter d'exposer le mot de passe de votre base de données source, le mot de passe</p>

Paramètre	Action
	<p>AWS SCT n'est pas stocké par défaut. Si vous fermez et rouvrez votre projet AWS SCT , vous êtes invité à entrer le mot de passe pour vous connecter à la base de données source, si nécessaire.</p>
Use SSL	<p>Choisissez cette option pour utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données. Fournissez les informations supplémentaires suivantes, le cas échéant, dans l'onglet SSL :</p> <ul style="list-style-type: none"> <li>• Vérifier le certificat du serveur : choisissez cette option pour vérifier le certificat du serveur à l'aide d'un trust store.</li> <li>• Trust store : un trust store que vous configurez dans les paramètres globaux.</li> <li>• Magasin de clés : magasin de clés que vous configurez dans les paramètres généraux.</li> </ul>
Enregistrer le mot de passe	<p>AWS SCT crée un coffre-fort sécurisé pour stocker les certificats SSL et les mots de passe de base de données. En activant cette option, vous pouvez enregistrer le mot de passe de la base de données et vous connecter rapidement à la base de données sans avoir à saisir le mot de passe.</p>
Trajectoire verticale du conducteur	<p>Entrez le chemin d'accès au pilote à utiliser pour vous connecter à la base de données source. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a>.</p> <p>Si vous stockez le chemin d'accès au pilote dans les paramètres globaux du projet, il ne s'affiche pas dans la boîte de dialogue de connexion. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des chemins des pilotes dans les paramètres globaux</a>.</p>

5. Choisissez Tester la connexion pour vérifier que AWS SCT vous pouvez vous connecter à votre base de données source.



6. Choisissez Connect pour vous connecter à votre base de données source.

## Paramètres de conversion de Vertica vers Amazon Redshift

Pour modifier les paramètres de conversion de Vertica vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Vertica, puis Vertica — Amazon Redshift. AWS SCT affiche tous les paramètres disponibles pour la conversion de Vertica vers Amazon Redshift.

Les paramètres de conversion de Vertica vers Amazon Redshift incluent des options pour AWS SCT les éléments suivants :

- Pour limiter le nombre de commentaires contenant des actions dans le code converti.

Pour Ajouter des commentaires dans le code converti pour les actions de gravité sélectionnée ou supérieure, choisissez la sévérité des actions. AWS SCT ajoute des commentaires dans le code converti pour les actions dont la gravité est sélectionnée ou supérieure.

Par exemple, pour réduire au maximum le nombre de commentaires dans votre code converti, choisissez Erreurs uniquement. Pour inclure les commentaires pour tous les éléments d'action de votre code converti, choisissez Tous les messages.

- Pour définir le nombre maximum de tables AWS SCT pouvant s'appliquer à votre cluster Amazon Redshift cible.

Pour le nombre maximum de tables pour le cluster Amazon Redshift cible, choisissez le nombre de tables que AWS SCT peuvent s'appliquer à votre cluster Amazon Redshift.

Amazon Redshift dispose de quotas qui limitent l'utilisation des tables pour différents types de nœud de cluster. Si vous choisissez Auto, AWS SCT détermine le nombre de tables à appliquer à votre cluster Amazon Redshift cible en fonction du type de nœud. Vous pouvez éventuellement choisir la valeur manuellement. Pour plus d'informations, consultez [Quotas et limites dans Amazon Redshift](#) dans le Guide de gestion Amazon Redshift.

AWS SCT convertit toutes vos tables sources, même s'il s'agit d'un volume supérieur à ce que votre cluster Amazon Redshift peut stocker. AWS SCT stocke le code converti dans votre projet et ne l'applique pas à la base de données cible. Si vous atteignez le quota de cluster Amazon Redshift pour les tables lorsque vous appliquez le code converti, un message d'avertissement s'affiche. AWS SCT applique également des tables à votre cluster Amazon Redshift cible jusqu'à ce que le nombre de tables atteigne la limite.

- Pour migrer des partitions de la table source vers des tables séparées dans Amazon Redshift. Pour ce faire, sélectionnez Utiliser la vue UNION ALL et entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour une seule table source.

Amazon Redshift ne prend pas en charge le partitionnement des tables. Pour imiter ce comportement et accélérer l'exécution des requêtes, AWS SCT vous pouvez migrer chaque partition de votre table source vers une table distincte dans Amazon Redshift. AWS SCT Crée ensuite une vue qui inclut les données de toutes ces tables.

AWS SCT détermine automatiquement le nombre de partitions de votre table source. Selon le type de partitionnement de votre table source, ce nombre peut dépasser le quota de tables que vous pouvez appliquer à votre cluster Amazon Redshift. Pour éviter d'atteindre ce quota, entrez le nombre maximum de tables cibles AWS SCT pouvant être créées pour les partitions d'une seule table source. L'option par défaut est 368 tables, ce qui représente une partition pendant 366 jours par an et deux tables pour NO RANGE et UNKNOWN partitions.

- Pour appliquer une compression aux colonnes d'un tableau Amazon Redshift. Pour ce faire, sélectionnez Utiliser le codage par compression.

AWS SCT attribue automatiquement un codage de compression aux colonnes à l'aide de l'algorithme Amazon Redshift par défaut. Pour plus d'informations, consultez la section [Encodages de compression](#) dans le manuel Amazon Redshift Database Developer Guide.

Par défaut, Amazon Redshift n'applique pas de compression aux colonnes définies comme clés de tri et de distribution. Vous pouvez modifier ce comportement et appliquer une compression à ces colonnes. Pour ce faire, sélectionnez Utiliser le codage de compression pour les colonnes KEY. Vous ne pouvez sélectionner cette option que lorsque vous sélectionnez l'option Utiliser le codage par compression.

## Paramètres d'optimisation de la conversion entre Vertica et Amazon Redshift

Pour modifier les paramètres d'optimisation de conversion de Vertica vers Amazon Redshift, choisissez Paramètres AWS SCT dans, puis sélectionnez Paramètres de conversion. Dans la liste supérieure, choisissez Vertica, puis Vertica — Amazon Redshift. Dans le volet de gauche, sélectionnez Stratégies d'optimisation. AWS SCT affiche les paramètres d'optimisation des conversions pour la conversion de Vertica vers Amazon Redshift.

Les paramètres AWS SCT d'optimisation de la conversion de Vertica vers Amazon Redshift incluent des options pour les éléments suivants :

- Pour travailler avec l'optimisation automatique des tables. Pour ce faire, sélectionnez Utiliser le réglage automatique des tables Amazon Redshift.

L'optimisation automatique des tables est un processus de réglage automatique dans Amazon Redshift qui optimise automatiquement la conception des tables. Pour plus d'informations, consultez la section [Utilisation de l'optimisation automatique des tables](#) dans le manuel Amazon Redshift Database Developer Guide.

Pour vous fier uniquement à l'optimisation automatique des tables, choisissez Aucune pour la stratégie initiale de sélection des clés.

- Pour choisir les clés de tri et de distribution selon votre stratégie.

Vous pouvez choisir des clés de tri et de distribution à l'aide des métadonnées Amazon Redshift, des informations statistiques ou des deux options. Pour Stratégie initiale de sélection des clés dans l'onglet Stratégies d'optimisation, choisissez l'une des options suivantes :

- Utiliser des métadonnées, ignorer les informations statistiques
- Ignorer les métadonnées, utiliser des informations statistiques
- Utiliser des métadonnées et des informations statistiques

Selon l'option que vous choisissez, vous pouvez sélectionner des stratégies d'optimisation. Entrez ensuite la valeur (0 à 100) pour chaque stratégie. Ces valeurs définissent le poids de chaque stratégie. À l'aide de ces valeurs de pondération, AWS SCT définit l'influence de chaque règle sur le choix des clés de distribution et de tri. Les valeurs par défaut sont basées sur les meilleures pratiques de AWS migration.

Vous pouvez définir la taille des petites tables dans le cadre de la stratégie Find small tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le définir comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

- Pour configurer les détails de la stratégie.

Outre la définition du poids pour chaque stratégie d'optimisation, vous pouvez configurer les paramètres d'optimisation. Pour ce faire, choisissez Optimisation des conversions.

- Pour le champ Limite des colonnes clés de tri, entrez le nombre maximum de colonnes dans la clé de tri.

- Pour une valeur de seuil asymétrique, entrez le pourcentage (0 à 100) d'une valeur asymétrique pour une colonne. AWS SCT exclut les colonnes dont la valeur d'inclinaison est supérieure au seuil de la liste des candidats pour la clé de distribution. AWS SCT définit la valeur asymétrique d'une colonne comme le pourcentage entre le nombre d'occurrences de la valeur la plus courante et le nombre total d'enregistrements.
- Pour les N principales requêtes de la table d'historique des requêtes, entrez le nombre (1 à 100) des requêtes les plus fréquemment utilisées à analyser.
- Pour Sélectionner un utilisateur de statistiques, choisissez l'utilisateur de base de données pour lequel vous souhaitez analyser les statistiques de requête.

Dans l'onglet Stratégies d'optimisation, vous pouvez également définir la taille des petites tables pour la stratégie Rechercher de petites tables. Pour Nombre minimum de lignes de tableau et Nombre maximum de lignes de tableau, entrez le nombre minimum et maximum de lignes dans un tableau pour le considérer comme un petit tableau. AWS SCT applique le style ALL de distribution aux petites tables. Dans ce cas, une copie de la table complète est distribuée à chaque nœud.

# Cartographie des types de données dans AWS Schema Conversion Tool

Vous pouvez ajouter plusieurs bases de données source et cible dans un même AWS SCT projet. Cela simplifie la gestion des projets lorsque vous migrez plusieurs bases de données vers différentes plateformes cibles.

Après avoir créé un nouveau projet et ajouté des bases de données source et cible, créez des règles de mappage. AWS SCT nécessite au moins une règle de mappage pour créer un rapport d'évaluation de la migration et convertir les schémas de base de données.

Une règle de mappage décrit une paire source-cible qui inclut un schéma de base de données source ou une base de données source et une plate-forme de base de données cible. Vous pouvez créer plusieurs règles de mappage dans un même AWS SCT projet. Utilisez les règles de mappage pour convertir chaque schéma de base de données source vers la plate-forme de base de données cible appropriée.

Pour modifier le nom de votre schéma dans le code converti, configurez une règle de migration. Par exemple, avec les règles de migration, vous pouvez renommer votre schéma, ajouter un préfixe aux noms d'objets, modifier le classement des colonnes ou modifier les types de données. Pour appliquer ces modifications à votre code converti, assurez-vous de créer des règles de migration avant de convertir votre schéma source. Pour de plus amples informations, veuillez consulter [Appliquer les règles de migration](#).

Vous pouvez créer des règles de mappage uniquement pour les paires de conversion de base de données prises en charge. Pour obtenir la liste des paires de conversion prises en charge, consultez [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#).

Si vous ouvrez un projet enregistré dans la AWS SCT version 1.0.655 ou antérieure, créez AWS SCT automatiquement des règles de mappage pour tous les schémas de base de données source vers la plate-forme de base de données cible. Pour ajouter d'autres plateformes de base de données cibles, supprimez les règles de mappage existantes, puis créez-en de nouvelles.

## Rubriques

- [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#)
- [Modification des mappages de types de données dans AWS Schema Conversion Tool](#)
- [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#)

- [Limites du mappage des types de données dans AWS Schema Conversion Tool](#)

## Cartographie de nouveaux types de données dans AWS Schema Conversion Tool

Vous pouvez créer plusieurs règles de mappage dans un même projet. AWS SCT enregistre les règles de mappage dans le cadre de votre projet. Votre projet étant ouvert, suivez la procédure ci-dessous pour ajouter une nouvelle règle de mappage.

Pour créer des règles de mappage

1. Dans le menu Affichage, choisissez Affichage cartographique.
2. Dans le panneau de gauche, choisissez un schéma ou une base de données à ajouter à la règle de mappage.
3. Dans le panneau de droite, choisissez une plate-forme de base de données cible pour le schéma source ou la base de données sélectionnés.

Vous pouvez choisir une plate-forme de base de données virtuelle comme cible. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Choisissez Créer un mappage.

AWS SCT ajoute cette nouvelle règle de mappage à la liste des mappages du serveur.

Ajoutez des règles de mappage pour toutes les paires de conversion. Pour créer un rapport d'évaluation ou convertir des schémas de base de données, choisissez Vue principale dans le menu Affichage.

AWS SCT met en évidence en gras tous les objets de schéma qui font partie d'une règle de mappage.

## Modification des mappages de types de données dans AWS Schema Conversion Tool

Vous pouvez filtrer ou supprimer les règles de mappage existantes et ajouter une nouvelle règle de mappage dans votre projet AWS Schema Conversion Tool (AWS SCT).

Lorsque vous créez une règle de mappage pour l'ensemble de la base de données source, AWS SCT crée une règle de mappage pour chaque schéma de base de données source. Pour les projets impliquant des dizaines de schémas ou même des bases de données, il peut être difficile de comprendre quelle cible est utilisée pour un schéma donné. Pour trouver rapidement une règle de mappage pour votre schéma, utilisez une ou plusieurs des options de filtre suivantes dans AWS SCT.

Pour filtrer les règles de mappage

1. Dans le menu Affichage, choisissez Affichage cartographique.
2. Pour les serveurs source, choisissez la base de données source.

Le filtre par défaut est Tous, ce qui signifie qu'il AWS SCT affiche les règles de mappage pour toutes les bases de données sources.

3. Pour Schéma source, entrez le nom du schéma source. Utilisez le pourcentage (%) comme caractère générique pour remplacer un certain nombre de symboles dans le nom du schéma.

Le filtre par défaut est le caractère générique %, ce qui signifie qu'il AWS SCT affiche les règles de mappage pour tous les noms de schéma de base de données source.

4. Pour Has migration rules, choisissez Yes pour afficher les règles de mappage pour lesquelles les règles de migration de données ont été créées. Choisissez Non pour afficher les règles de mappage qui n'ont pas de règles de migration de données. Pour de plus amples informations, veuillez consulter [Création de règles de migration de données dans AWS SCT](#).

Le filtre par défaut est Tous, ce qui signifie qu'il AWS SCT affiche toutes les règles de mappage.

5. Pour les serveurs cibles, choisissez la base de données cible.

Le filtre par défaut est Tous, ce qui signifie qu'il AWS SCT affiche les règles de mappage pour toutes les bases de données cibles.

Votre projet étant ouvert, suivez la procédure ci-dessous pour supprimer une règle de mappage. Pour plus d'informations sur l'ajout de règles de mappage, consultez [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).

Pour supprimer des règles de mappage

1. Dans le menu Affichage, choisissez Affichage cartographique.
2. Pour les mappages de serveurs, choisissez les règles de mappage à supprimer.
3. Choisissez Supprimer les mappages sélectionnés.

AWS SCT supprime les règles de mappage sélectionnées.

## Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool

Vous pouvez voir comment AWS SCT convertir le schéma de votre base de données source en n'importe quelle plate-forme de base de données cible prise en charge. Pour ce faire, vous n'avez pas besoin de vous connecter à une base de données cible existante. Vous pouvez plutôt choisir une plate-forme de base de données cible virtuelle dans le panneau de droite lorsque vous créez une règle de mappage. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#). Assurez-vous de développer les serveurs, les clusters NoSQL et les nœuds ETL dans le panneau de droite pour afficher la liste des plateformes de base de données cibles virtuelles.

AWS SCT prend en charge les plateformes de base de données cibles virtuelles suivantes :

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift et AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish for Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Si vous utilisez Babelfish pour Aurora PostgreSQL comme plate-forme de base de données cible, vous ne pouvez créer qu'un rapport d'évaluation de la migration de base de données. Pour de plus amples informations, veuillez consulter [the section called "Rapport d'évaluation"](#).



Si vous utilisez une plate-forme de base de données cible virtuelle, vous pouvez enregistrer le code converti dans un fichier. Pour de plus amples informations, veuillez consulter [the section called “Sauvegarde de votre schéma converti”](#).

## Limites du mappage des types de données dans AWS Schema Conversion Tool

Les limites suivantes s'appliquent lors de la conversion de schémas utilisant plusieurs serveurs dans un même AWS SCT projet :

- Vous ne pouvez ajouter le même serveur à un projet qu'une seule fois.
- Vous ne pouvez pas mapper des schémas de serveur à un schéma cible spécifique, mais uniquement à un serveur cible. AWS SCT crée le schéma cible lors de la conversion.
- Vous ne pouvez pas mapper des objets source de niveau inférieur sur le serveur cible.
- Vous ne pouvez mapper un schéma source qu'à un seul serveur cible dans un projet.
- Assurez-vous de mapper une source à un serveur cible pour créer un rapport d'évaluation, convertir des schémas ou extraire des données.

# Utilisation des rapports dans le AWS Schema Conversion Tool

Lorsque vous envisagez une conversion de base de données, il est utile de créer des rapports pour vous aider à comprendre ce que cela implique. Vous pouvez créer des rapports à l'aide de AWS Schema Conversion Tool.

Vous pouvez l'utiliser AWS SCT pour créer un rapport d'évaluation de la migration de base de données. Ce rapport comporte un résumé de vos tâches de conversion de schéma et les détails des éléments qui ne peuvent pas être convertis automatiquement vers votre base de données cible. Vous pouvez utiliser ce rapport pour évaluer la partie du projet qui peut être réalisée à l'aide et AWS SCT les autres éléments dont vous avez besoin pour effectuer la conversion. Pour créer un rapport d'évaluation, utilisez Create Report (Créer un rapport) dans le menu contextuel (clic droit) de la base de données dans AWS SCT.

## Rubriques

- [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#)

## À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool

Une partie importante du document AWS Schema Conversion Tool est le rapport d'évaluation qu'il génère pour estimer la complexité de la conversion de votre schéma. Ce rapport d'évaluation de la migration de base de données résume toutes les tâches de conversion de schéma et détaille les mesures à prendre pour le schéma qui ne peuvent pas être convertis vers le moteur de base de données de votre instance de base de données cible. Vous pouvez afficher le rapport dans l'application ou l'exporter sous forme de valeur séparée par des virgules (CSV) ou de fichier PDF.

Si vous ajoutez plusieurs bases de données source et cible dans un même projet, AWS SCT regroupe les rapports de toutes les paires de conversions en un seul rapport d'évaluation de la migration de base de données.

Vous pouvez utiliser des plateformes de base de données cibles virtuelles pour générer un rapport d'évaluation et comprendre la complexité de la migration vers une plate-forme de base de données sélectionnée. Dans ce cas, vous n'avez pas besoin de vous connecter à votre plate-forme de base de données cible. Par exemple, vous pouvez utiliser Babelfish pour Aurora PostgreSQL comme

plate-forme de base de données cible virtuelle pour créer un rapport d'évaluation de la migration de base de données. Pour plus d'informations sur les plateformes de base de données cibles virtuelles, consultez [the section called “Cartographie de cibles virtuelles”](#).

Le rapport d'évaluation de migration comprend les éléments suivants :

- Une synthèse
- Une évaluation de licence
- Un support cloud, indiquant tous les fonctions de la base de données source non disponibles sur la cible.
- Des recommandations, y compris la conversion d'objets serveur, des suggestions de sauvegarde et les modifications de serveur liées

Ce rapport comprend également des estimations des efforts qui devront être fournis pour écrire le code équivalent pour votre instance de base de données cible qui ne peut pas être converti automatiquement.

Si vous avez l' AWS SCT habitude de migrer votre schéma existant vers une instance de base de données Amazon RDS, vous pouvez utiliser le rapport pour vous aider à analyser les exigences relatives à la migration vers le AWS cloud et à modifier votre type de licence.

## Rubriques

- [Création d'un rapport d'évaluation dans AWS Schema Conversion Tool](#)
- [Afficher un rapport d'évaluation dans AWS Schema Conversion Tool](#)
- [Sauvegarde du rapport d'évaluation dans AWS Schema Conversion Tool](#)
- [Configuration d'un rapport d'évaluation dans AWS Schema Conversion Tool](#)
- [Création d'un rapport d'évaluation multiserveur dans AWS Schema Conversion Tool](#)

## Création d'un rapport d'évaluation dans AWS Schema Conversion Tool

Utilisez la procédure suivante pour créer un rapport d'évaluation de la migration de base de données.

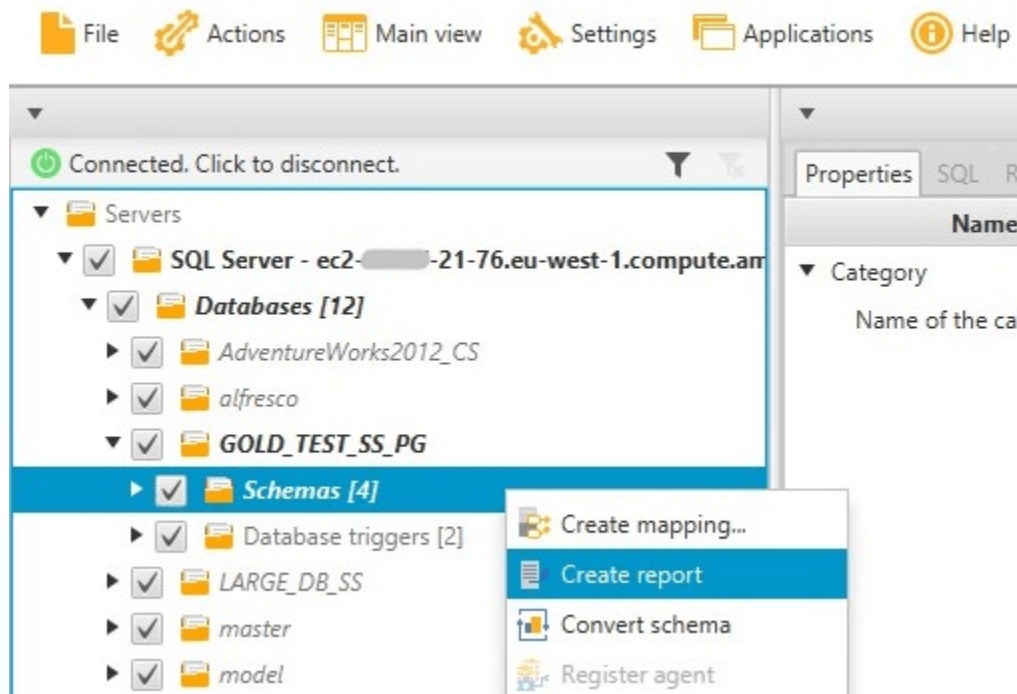
Pour créer un rapport d'évaluation de migration de base de données

1. Assurez-vous d'avoir créé une règle de mappage pour le schéma de base de données source afin de créer un rapport d'évaluation pour. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).

2. Dans le menu Affichage, choisissez Vue principale.
3. Dans le panneau de gauche qui affiche le schéma de votre base de données source, choisissez un objet de schéma pour lequel créer un rapport d'évaluation. Pour inclure plusieurs schémas de base de données dans le rapport, choisissez le nœud parent, par exemple Schémas.

Assurez-vous d'avoir coché les cases correspondant à tous les objets de schéma pour lesquels créer un rapport d'évaluation.

4. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Créer un rapport.



## Afficher un rapport d'évaluation dans AWS Schema Conversion Tool

Après avoir créé un rapport d'évaluation, la vue de ce dernier s'ouvre et affiche les onglets suivants :

- Récapitulatif
- Action Items (Éléments d'action)

L'onglet Summary (Résumé) affiche les éléments qui ont été convertis automatiquement ou qui n'ont pas été convertis.

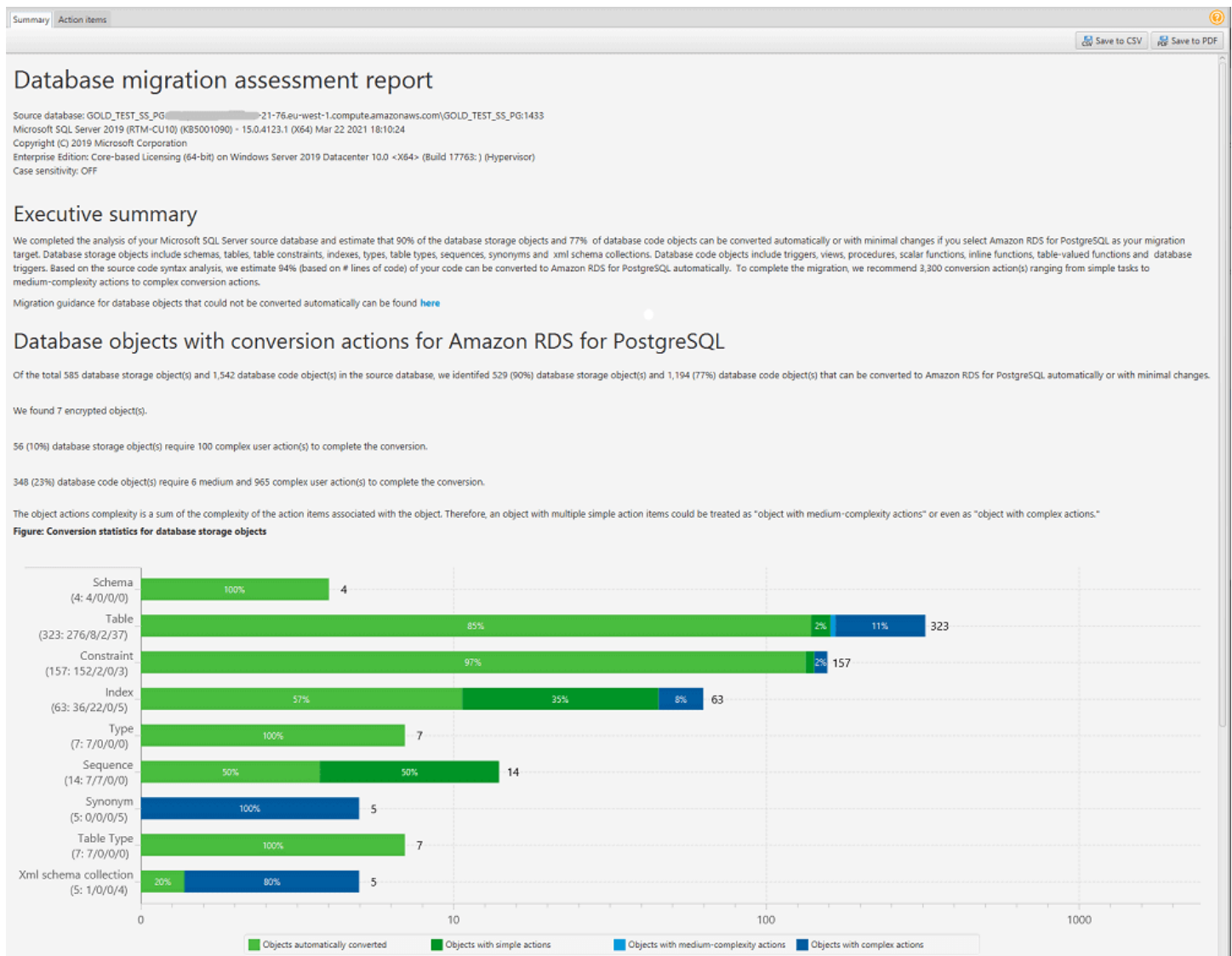
L'onglet Action Items (Éléments d'action) affiche les éléments qui ne peuvent pas être convertis automatiquement, ainsi que des recommandations sur ce qu'il convient de faire dans ce cas.

## Rubriques

- [Récapitulatif du rapport d'évaluation](#)
- [Éléments d'action du rapport d'évaluation](#)
- [Message d'avertissement du rapport d'évaluation](#)

## Récapitulatif du rapport d'évaluation

L'onglet Summary affiche un récapitulatif du rapport d'évaluation de migration de base de données. Il indique les éléments qui ont été convertis automatiquement et ceux qui ne l'ont pas été.



En ce qui concerne les éléments du schéma qui ne peuvent pas être convertis automatiquement pour le moteur de base de données cible, ce récapitulatif comprend une estimation de l'effort nécessaire

pour créer des éléments de schéma dans votre instance DB cible équivalents à ceux présents dans votre source.

Ce rapport classe le temps estimé pour convertir ces éléments de schéma de la façon suivante :

- Simple — Actions pouvant être effectuées en moins de deux heures.
- Moyen — Actions plus complexes pouvant être réalisées en deux à six heures.
- Important — Actions très complexes dont la réalisation prend plus de six heures.

La section Évaluation des licences et support cloud contient des informations sur le déplacement de votre schéma de base de données sur site existant vers une instance de base de données Amazon RDS exécutant le même moteur. Par exemple, si vous voulez modifier les types de licence, cette section du rapport vous indique quelles fonctions de votre base de données actuelle doivent être supprimées.

**License evaluation**  
Our analysis shows that current schema uses the following Enterprise Edition features unavailable in Standard Edition.

Feature	Description
Database In-Memory	Oracle Database In-Memory optimizes both analytics and mixed workload OLTP, delivering outstanding performance for transactions while simultaneously supporting real-time analytics, business intelligence, and reports.
Materialized View Query Rewrite	Oracle Database employs an extremely powerful process called query rewrite to quickly answer the query using materialized views.
Partitioning	Partitioning is powerful functionality that allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity.
Oracle Advanced Security/TDE	Oracle Advanced Security provides two important preventive controls to protect sensitive data at the source: encryption and redaction. Together, these two controls form the foundation of Oracle's defense-in-depth, multi-layered database security solution.

If you choose Standard Edition as your migration target, remove dependencies on these features.

**Cloud support**  
Our analysis shows that your current schema uses the following features that require configuration steps in Amazon RDS for Oracle.

Feature	Description
Locator	Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. Please read prerequisites and configuration steps in the next article: <a href="#">Oracle Locator</a> .
Spatial	Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial data in an Oracle database. Please read prerequisites and configuration steps in the next article: <a href="#">Oracle Spatial</a> .
Oracle XML DB	Oracle XML DB provides full support for all of the key XML standards, including XML Namespaces, DOM, XQuery, SQL/XML and XSLT. Amazon RDS for Oracle supports XML DB feature without the XML DB Protocol Server. Please read prerequisites and configuration steps in the next article: <a href="#">Oracle XML DB option</a> .

If choose Amazon RDS for Oracle as your migration target, please follow the abovementioned steps to continue to use these features on the target database after migration completes.

## Éléments d'action du rapport d'évaluation

L'affichage de rapport d'évaluation comprend également un onglet Action Items. Cet onglet contient une liste d'éléments qui ne peuvent pas être convertis automatiquement vers le moteur de base de données de votre instance de base de données Amazon RDS cible. Si vous sélectionnez un élément d'action dans la liste, AWS SCT met en évidence l'élément de votre schéma auquel s'applique l'élément d'action.

Le rapport contient également des recommandations sur la manière de convertir manuellement l'élément de schéma. Par exemple, après l'exécution de l'évaluation, des rapports détaillés relatifs à la base de données/au schéma indiquent les efforts requis pour concevoir et mettre en œuvre les

recommandations relatives à la conversion des éléments d'action. Pour plus d'informations sur la façon de gérer les conversions manuelles, consultez [Conversion de schémas à l'aide d'AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, schemas, and tables. The main pane shows a list of issues with their recommended actions and occurrence counts. Below the issues, a SQL query is shown for a stored procedure.

**Issues:**

- Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
- Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes. Number of occurrences: 2
- Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary. Number of occurrences: 1
- Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable. Number of occurrences: 1
- Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy. Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
- Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually. Number of occurrences: 3
- Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion. Number of occurrences: 1
- Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function. Number of occurrences: 12

**SQL Query:**

```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint(4) readonly
3   , @posFlags bigint - 0
4   , @posFlagsMask bigint - 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask ) | @posFlags
8 from Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0

```

## Message d'avertissement du rapport d'évaluation

Pour évaluer la complexité de la conversion vers un autre moteur de base de données, AWS SCT vous devez accéder aux objets de votre base de données source. Lorsque le SCT ne peut pas effectuer d'évaluation en raison de problèmes rencontrés lors de l'analyse, un message d'avertissement est émis indiquant que le pourcentage de conversion global est réduit.

### Warning!

We found that your source database may be configured not in correct way or you have not enough privileges for reading all necessary metadata. Please check your configuration and run report again. For more details please review [help documentation](#).

List of Action Items to review:

- Issue 9997** Unable to resolve objects (number of occurrences: 3)  
Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.

Les raisons pour lesquelles des problèmes AWS SCT peuvent survenir lors de la numérisation sont les suivantes :

- Le compte utilisateur connecté à la base de données n'a pas accès à tous les objets nécessaires.
- Un objet cité dans le schéma n'existe plus dans la base de données.
- Le SCT essaie d'évaluer un objet chiffré.

Pour plus d'informations sur les autorisations et privilèges de sécurité requis par le SCT pour votre base de données, consultez [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#) la section de ce guide consacrée à la base de données source appropriée.

## Sauvegarde du rapport d'évaluation dans AWS Schema Conversion Tool

Après avoir [créé un rapport d'évaluation de la migration de base](#) de données, vous pouvez enregistrer une copie locale du rapport d'évaluation de la migration de base de données sous forme de fichier PDF ou de fichier CSV (valeurs séparées par des virgules).

Pour enregistrer un rapport d'évaluation de migration de base de données au format PDF

1. Dans le menu supérieur, choisissez Afficher, puis Afficher le rapport d'évaluation.
2. Choisissez l'onglet Summary.
3. Choisissez Enregistrer au format PDF en haut à droite.

Pour enregistrer un rapport d'évaluation de la migration de base de données sous forme de fichier CSV

1. Dans le menu supérieur, choisissez Afficher, puis Afficher le rapport d'évaluation.
2. Choisissez l'onglet Summary.
3. Choisissez Enregistrer au format CSV en haut à droite.

Le fichier PDF contient les informations sur le récapitulatif et les éléments d'action, comme illustré dans l'exemple suivant.



## Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

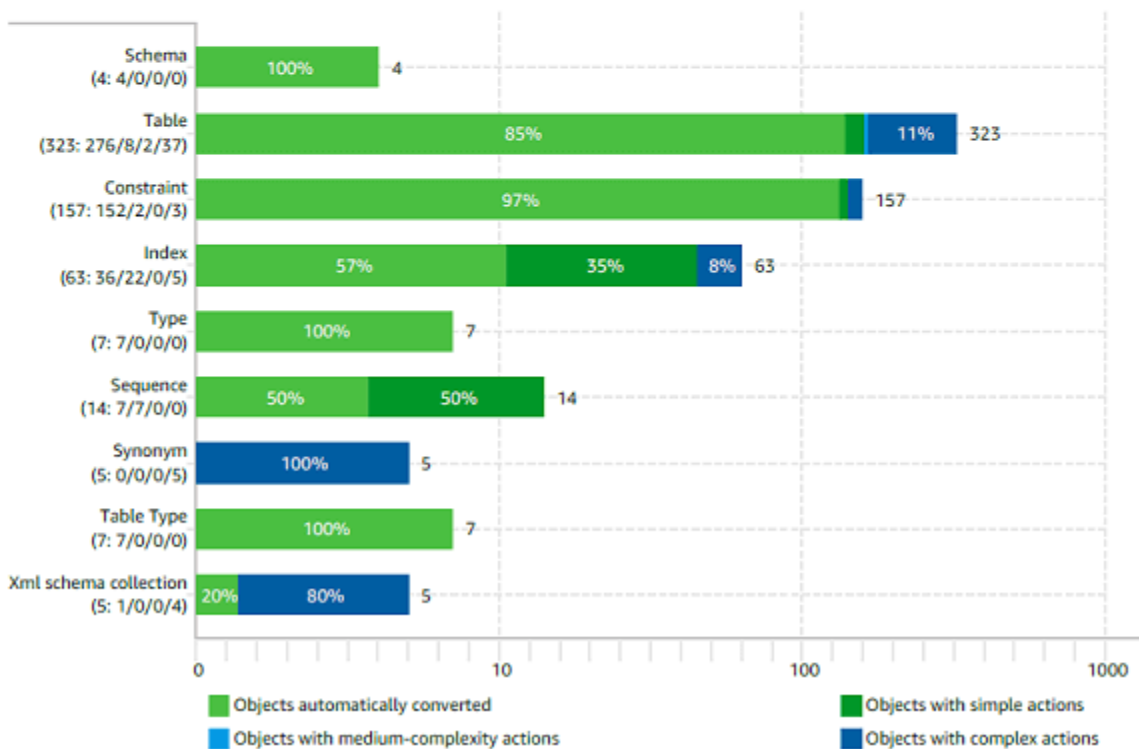
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects



Lorsque vous choisissez l'option Enregistrer au format CSV, AWS SCT trois fichiers CSV sont créés.

Le premier fichier CSV contient les informations suivantes sur les éléments d'action :

- Catégorie
- Occurrence : nom du fichier, numéro de ligne et position de l'élément
- Numéro de l'élément d'action
- Sujet
- Groupe

- Description
- Références de documentation
- Action recommandée
- Complexité estimée

Le deuxième fichier CSV inclut le `Action_Items_Summary` suffixe dans son nom et contient les informations sur le nombre d'occurrences de toutes les actions.

Dans l'exemple suivant, les valeurs de la colonne d'effort de la courbe d'apprentissage indiquent l'effort nécessaire pour concevoir une approche permettant de convertir chaque élément d'action. Les valeurs de la colonne `Effort` pour convertir une occurrence de l'élément d'action indiquent l'effort nécessaire pour convertir chaque élément d'action, conformément à l'approche conçue. Les valeurs utilisées pour indiquer le niveau d'effort requis sont basées sur une échelle pondérée, allant de faible (minimum) à élevé (maximum).

Schema	Action item	Number of occurrences	Learning curve efforts	Efforts to convert an occurrence of the action item
TEST.dbo	609	1	8	0.3
TEST.dbo	681	2	0.1	0.1
TEST.dbo	690	1	40	40
TEST.dbo	794	1	0	0.01
TEST.dbo	811	12	40	8
TEST.dbo	826	1	0	0.1
TEST.dbo	844	8	8	0.5
TEST.dbo	9997	3	0	0.3

Le troisième fichier CSV inclut `Summary` dans son nom et contient le résumé suivant :

- Catégorie
- Nombre d'objets
- Objets automatiquement convertis
- Objets avec des actions simples
- Objets avec des actions de complexité moyenne
- Objets avec des actions complexes
- Nombre total de lignes de code

## Configuration d'un rapport d'évaluation dans AWS Schema Conversion Tool

Vous pouvez configurer le nombre de détails AWS SCT inclus dans les rapports d'évaluation.

## Pour configurer un rapport d'évaluation de la migration de base de données

1. Dans le menu Paramètres, choisissez Paramètres généraux, puis sélectionnez Rapport d'évaluation.
2. Pour les occurrences d'actions, sélectionnez les cinq premiers problèmes uniquement pour limiter le nombre d'actions d'un même type dans le rapport d'évaluation. Choisissez Tous les problèmes pour inclure tous les éléments d'action de chaque type dans le rapport d'évaluation.
3. Pour les fichiers analysés par script SQL, sélectionnez Ne répertorier que des **X** fichiers pour limiter le nombre de fichiers de script SQL dans le rapport d'évaluation à **X**. Entrez le nombre de fichiers. Choisissez Lister tous les fichiers analysés pour inclure tous les fichiers de script SQL dans le rapport d'évaluation.
4. Sélectionnez Ouvrir les rapports après l'enregistrement pour ouvrir automatiquement le fichier après avoir enregistré une copie locale du rapport d'évaluation de la migration de la base de données. Pour de plus amples informations, veuillez consulter

---

[Après avoir créé un rapport d'évaluation de la migration de base de données, vous pouvez enregistrer une copie locale du rapport d'évaluation de la migration de base de données sous forme de fichier PDF ou de fichier CSV \(valeurs séparées par des virgules\).](#)

---

[Pour enregistrer un rapport d'évaluation de migration de base de données au format PDF](#)

- 
1. Dans le menu supérieur, choisissez Afficher, puis Afficher le rapport d'évaluation.
  2. Choisissez l'onglet Summary.
  3. Choisissez Enregistrer au format PDF en haut à droite.
- 

[Pour enregistrer un rapport d'évaluation de la migration de base de données sous forme de](#)

---

[fichier CSV](#)

---

1. Dans le menu supérieur, choisissez Afficher, puis Afficher le rapport d'évaluation.
  2. Choisissez l'onglet Summary.
  3. Choisissez Enregistrer au format CSV en haut à droite.
- 

[Le fichier PDF contient les informations sur le récapitulatif et les éléments d'action, comme illustré dans l'exemple suivant.](#)

---

## Database objects with conversion actions for Amazon RDS for PostgreSQL

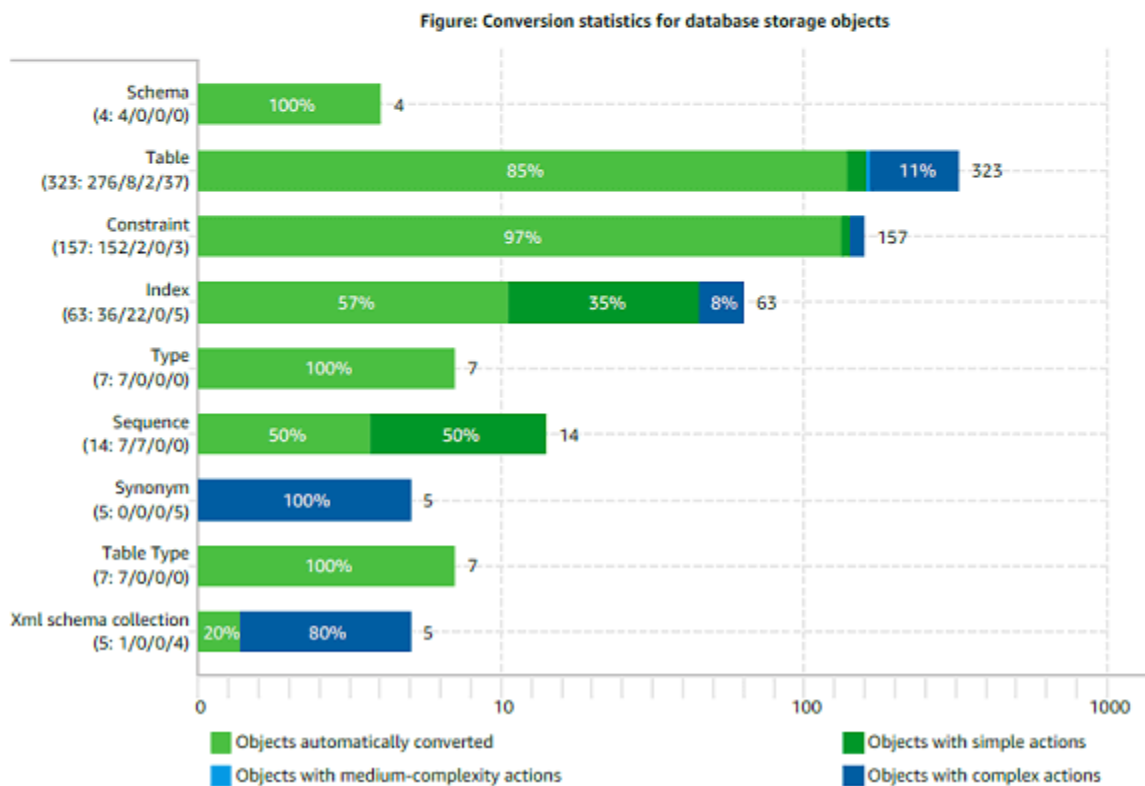
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."



Lorsque vous choisissez l'option Enregistrer au format CSV, AWS SCT trois fichiers CSV sont créés.

Le premier fichier CSV contient les informations suivantes sur les éléments d'action :

- Catégorie
- Occurrence : nom du fichier, numéro de ligne et position de l'élément
- Numéro de l'élément d'action
- Sujet

- Groupe

---

- Description

---

- Références de documentation

---

- Action recommandée

---

- Complexité estimée

---

Le deuxième fichier CSV inclut le `Action_Items_Summary` suffixe dans son nom et contient les informations sur le nombre d'occurrences de toutes les actions.

---

Dans l'exemple suivant, les valeurs de la colonne d'effort de la courbe d'apprentissage indiquent l'effort nécessaire pour concevoir une approche permettant de convertir chaque élément d'action. Les valeurs de la colonne Effort pour convertir une occurrence de l'élément d'action indiquent l'effort nécessaire pour convertir chaque élément d'action, conformément à l'approche conçue. Les valeurs utilisées pour indiquer le niveau d'effort requis sont basées sur une échelle pondérée, allant de faible (minimum) à élevé (maximum).

Schema	Action item	Number of occurrences	Learning curve efforts	Efforts to convert an occurrence of the action item
TEST.dbo	609	1	8	0.3
TEST.dbo	681	2	0.1	0.1
TEST.dbo	690	1	40	40
TEST.dbo	794	1	0	0.01
TEST.dbo	811	12	40	8
TEST.dbo	826	1	0	0.1
TEST.dbo	844	8	8	0.5
TEST.dbo	9997	3	0	0.3

---

Le troisième fichier CSV inclut `Summary` dans son nom et contient le résumé suivant :

---

- Catégorie

---

- Nombre d'objets

---

- Objets automatiquement convertis

---

- Objets avec des actions simples

---

- Objets avec des actions de complexité moyenne

---

- Objets avec des actions complexes

---

- Nombre total de lignes de code

---

## Création d'un rapport d'évaluation multiserveur dans AWS Schema Conversion Tool

Pour déterminer la meilleure orientation cible pour votre environnement global, créez un rapport d'évaluation multiserveur.

Un rapport d'évaluation multiserveur évalue plusieurs serveurs en fonction des entrées que vous fournissez pour chaque définition de schéma que vous souhaitez évaluer. La définition de votre schéma contient les paramètres de connexion au serveur de base de données et le nom complet de chaque schéma. Après avoir évalué chaque schéma, AWS SCT produit un rapport d'évaluation récapitulatif et agrégé pour la migration de bases de données sur vos multiples serveurs. Ce rapport indique la complexité estimée pour chaque cible de migration possible.

Vous pouvez l'utiliser AWS SCT pour créer un rapport d'évaluation multiserveur pour les bases de données source et cible suivantes.

Base de données source	Base de données cible
Amazon Redshift	Amazon Redshift
Base de données Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Analyses Azure Synapse	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM DB2 pour z/OS	Édition compatible avec Amazon Aurora MySQL (Aurora MySQL), édition compatible avec Amazon Aurora PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL
IBM Db2 LUW	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, Babelfish pour Aurora PostgreSQL

Base de données source	Base de données cible
	L, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Snowflake	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

## Réalisation d'une évaluation multiserveur

Utilisez la procédure suivante pour effectuer une évaluation multiserveur avec AWS SCT. Il n'est pas nécessaire de créer un nouveau projet AWS SCT pour effectuer une évaluation multiserveur. Avant de commencer, assurez-vous d'avoir préparé un fichier de valeurs séparées par des virgules (CSV) avec les paramètres de connexion à la base de données. Assurez-vous également d'avoir installé tous les pilotes de base de données requis et de définir l'emplacement des pilotes dans les AWS SCT paramètres. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

Pour effectuer une évaluation multiserveur et créer un rapport de synthèse agrégé

1. Dans AWS SCT, choisissez Fichier, Nouvelle évaluation multiserveur. La boîte de dialogue Nouvelle évaluation multiserveur s'ouvre.

**New multiserver assessment** ✕

Enter the project name, location to store reports and project files, and location of your connections file. ?

Project name

Location

Connections file

[Download a connections file example](#)

Create AWS SCT projects for each source database

Add mapping rules to these projects and save conversion statistics for offline use

2. Choisissez Télécharger un exemple de fichier de connexions pour télécharger un modèle vide de fichier CSV contenant les paramètres de connexion à la base de données.
3. Entrez des valeurs pour le nom du projet, l'emplacement (pour stocker les rapports) et le fichier de connexions (un fichier CSV).
4. Choisissez Créer des AWS SCT projets pour chaque base de données source afin de créer automatiquement des projets de migration après avoir généré le rapport d'évaluation.
5. Lorsque l'option Créer AWS SCT des projets pour chaque base de données source est activée, vous pouvez choisir Ajouter des règles de mappage à ces projets et enregistrer les statistiques de conversion pour une utilisation hors ligne. Dans ce cas, AWS SCT ajoutera des règles de mappage à chaque projet et enregistrera les métadonnées de la base de données source dans le projet. Pour de plus amples informations, veuillez consulter [Utilisation du mode hors ligne dans AWS Schema Conversion Tool](#).
6. Cliquez sur Exécuter.

Une barre de progression apparaît pour indiquer le rythme d'évaluation de la base de données. Le nombre de moteurs cibles peut affecter le temps d'exécution de l'évaluation.

7. Choisissez Oui si le message suivant s'affiche : L'analyse complète de tous les serveurs de base de données peut prendre un certain temps. Voulez-vous continuer ?

Lorsque le rapport d'évaluation multiserveur est terminé, un écran apparaît pour l'indiquer.

8. Choisissez Ouvrir le rapport pour afficher le rapport d'évaluation récapitulatif agrégé.



AWS SCT Génère par défaut un rapport agrégé pour toutes les bases de données sources et un rapport d'évaluation détaillé pour chaque nom de schéma dans une base de données source. Pour de plus amples informations, veuillez consulter [Localisation et affichage des rapports](#).

Lorsque l'option Créer AWS SCT des projets pour chaque base de données source est activée, AWS SCT crée un projet vide pour chaque base de données source. AWS SCT crée également des rapports d'évaluation comme décrit précédemment. Après avoir analysé ces rapports d'évaluation et choisi la destination de migration pour chaque base de données source, ajoutez des bases de données cibles à ces projets vides.

Lorsque l'option Ajouter des règles de mappage à ces projets et enregistrer les statistiques de conversion pour une utilisation hors ligne est activée, un projet est AWS SCT créé pour chaque base de données source. Ces projets incluent les informations suivantes :

- Votre base de données source et une plate-forme de base de données cible virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).
- Une règle de mappage pour cette paire source-cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données](#).
- Rapport d'évaluation de la migration de base de données pour cette paire source-cible.
- Les métadonnées du schéma source, qui vous permettent d'utiliser ce AWS SCT projet en mode hors ligne. Pour de plus amples informations, veuillez consulter [Utilisation du mode hors ligne dans AWS Schema Conversion Tool](#).

## Préparation d'un fichier CSV d'entrée

Pour fournir des paramètres de connexion en entrée pour le rapport d'évaluation multiserveur, utilisez un fichier CSV comme indiqué dans l'exemple suivant.

```
Name,Description,Secret Manager Key,Server IP,Port,Service Name,Database name,BigQuery
path,Source Engine,Schema Names,Use Windows Authentication,Login,Password,Use
SSL,Trust store,Key store,SSL authentication,Target Engines
Sales,,192.0.2.0,1521,pdb,,ORACLE,Q4_2021;FY_2021,,user,password,,,,,POSTGRESQL;AURORA_POSTGR
Marketing,,,ec2-a-b-c-d.eu-
west-1.compute.amazonaws.com,1433,,target_audience,,MSSQL,customers.dbo,,user,password,,,,,AURO
HR,,,192.0.2.0,1433,,employees,,MSSQL,employees.%,true,,,,,,AURORA_POSTGRESQL
Customers,,secret-name,,,,,MYSQL,customers,,,,,,AURORA_POSTGRESQL
Analytics,,,198.51.100.0,8195,,STATISTICS,,DB2LUW,BI_REPORTS,,user,password,,,,,POSTGRESQL
```

```
Products,,,203.0.113.0,8194,,,TERADATA,new_products,,user,password,,,,,REDSHIFT
```

L'exemple précédent utilise un point-virgule pour séparer les deux noms de schéma de la base de données. Sales Il utilise également un point-virgule pour séparer les deux plateformes de migration de base de données cibles pour la base de données. Sales

L'exemple précédent utilise également la connexion AWS Secrets Manager à la Customers base de données et l'authentification Windows pour la connexion à la HR base de données.

Vous pouvez créer un nouveau fichier CSV ou télécharger un modèle de fichier CSV depuis AWS SCT et renseigner les informations requises. Assurez-vous que la première ligne de votre fichier CSV inclut les mêmes noms de colonnes que ceux illustrés dans l'exemple précédent.

Pour télécharger un modèle du fichier CSV d'entrée

1. Démarrez AWS SCT.
2. Choisissez Fichier, puis choisissez Nouvelle évaluation multiserveur.
3. Choisissez Télécharger un exemple de fichier de connexions.

Assurez-vous que votre fichier CSV inclut les valeurs suivantes, fournies par le modèle :

- Nom : étiquette de texte qui permet d'identifier votre base de données. AWS SCT affiche cette étiquette de texte dans le rapport d'évaluation.
- Description — Valeur facultative, dans laquelle vous pouvez fournir des informations supplémentaires sur la base de données.
- Clé de gestion secrète : nom du secret qui stocke les informations d'identification de votre base de données dans le AWS Secrets Manager. Pour utiliser Secrets Manager, assurez-vous de stocker AWS les profils dans AWS SCT. Pour de plus amples informations, veuillez consulter [Configuration AWS Secrets Manager dans le AWS Schema Conversion Tool](#).

#### Important

AWS SCT ignore le paramètre Secret Manager Key si vous incluez les paramètres IP du serveur, port, identifiant et mot de passe dans le fichier d'entrée.

- IP du serveur : nom ou adresse IP du service de noms de domaine (DNS) de votre serveur de base de données source.
- Port : port utilisé pour se connecter à votre serveur de base de données source.

- Nom du service : si vous utilisez un nom de service pour vous connecter à votre base de données Oracle, il s'agit du nom du service Oracle auquel vous souhaitez vous connecter.
- Nom de la base de données : nom de la base de données. Pour les bases de données Oracle, utilisez l'identifiant système Oracle (SID).
- BigQuery path : chemin d'accès au fichier clé du compte de service pour votre BigQuery base de données source. Pour plus d'informations sur la création de ce fichier, consultez [Privilèges pour BigQuery en tant que source](#).
- Moteur source : type de votre base de données source. Utilisez l'une des valeurs suivantes :
  - AZURE\_MSSQL pour une base de données Azure SQL.
  - AZURE\_SYMAPSE pour une base de données Azure Synapse Analytics.
  - GOOGLE\_BIGQUERY pour une base de données. BigQuery
  - DB2ZOS pour une base de données IBM Db2 for z/OS.
  - DB2LUW pour une base de données IBM DB2 LUW.
  - GREENPLUM pour une base de données Greenplum.
  - MSSQL pour une base de données Microsoft SQL Server.
  - MYSQL pour une base de données MySQL.
  - NETEZZA pour une base de données Netezza.
  - ORACLE pour une base de données Oracle.
  - POSTGRESQL pour une base de données PostgreSQL.
  - REDSHIFT pour une base de données Amazon Redshift.
  - SNOWFLAKE pour une base de données Snowflake.
  - SYBASE\_ASE pour une base de données SAP ASE.
  - TERADATA pour une base de données Teradata.
  - VERTICA pour une base de données Vertica.
- Noms des schémas : noms des schémas de base de données à inclure dans le rapport d'évaluation.

Pour Azure SQL Database, Azure Synapse Analytics BigQuery, Netezza, SAP ASE, Snowflake et SQL Server, utilisez le format suivant pour le nom du schéma :

*db\_name.schema\_name*

Remplacez *schema\_name* par le nom du schéma source.

Placez les noms de base de données ou de schéma qui incluent un point entre guillemets doubles, comme indiqué ci-dessous : "database.name"."schema.name".

Séparez plusieurs noms de schéma en utilisant des points-virgules comme indiqué ci-dessous :

Schema1;Schema2

Les noms de base de données et de schéma distinguent les majuscules et minuscules.

Utilisez le pourcentage (%) comme caractère générique pour remplacer un certain nombre de symboles dans le nom de la base de données ou du schéma. L'exemple précédent utilise le pourcentage (%) comme caractère générique pour inclure tous les schémas de la employees base de données dans le rapport d'évaluation.

- Utiliser l'authentification Windows : si vous utilisez l'authentification Windows pour vous connecter à votre base de données Microsoft SQL Server, entrez true. Pour de plus amples informations, veuillez consulter [Utilisation de l'authentification Windows lors de l'utilisation de Microsoft SQL Server comme source](#).
- Login : nom d'utilisateur permettant de se connecter à votre serveur de base de données source.
- Mot de passe : mot de passe pour vous connecter à votre serveur de base de données source.
- Utiliser le protocole SSL : si vous utilisez le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données source, entrez true.
- Trust store : le trust store à utiliser pour votre connexion SSL.
- Magasin de clés — Le magasin de clés à utiliser pour votre connexion SSL.
- Authentification SSL — Si vous utilisez l'authentification SSL par certificat, entrez true.
- Moteurs cibles : plateformes de base de données cibles. Utilisez les valeurs suivantes pour spécifier une ou plusieurs cibles dans le rapport d'évaluation :
  - AURORA\_MYSQL pour une base de données compatible avec Aurora MySQL.
  - AURORA\_POSTGRESQL pour une base de données compatible avec Aurora PostgreSQL.
  - BABELFISH pour une base de données PostgreSQL Babelfish pour Aurora.
  - MARIA\_DB pour une base de données MariaDB.
  - MSSQL pour une base de données Microsoft SQL Server.
  - MYSQL pour une base de données MySQL.
  - ORACLE pour une base de données Oracle.

- POSTGRESQL pour une base de données PostgreSQL.
- REDSHIFT pour une base de données Amazon Redshift.

Séparez plusieurs cibles en utilisant des points-virgules comme celui-ci : `MYSQL ; MARIA_DB` Le nombre de cibles influe sur le temps nécessaire à l'exécution de l'évaluation.

## Localisation et affichage des rapports

L'évaluation multiserveur génère deux types de rapports :

- Rapport agrégé de toutes les bases de données sources.
- Rapport d'évaluation détaillé des bases de données cibles pour chaque nom de schéma dans une base de données source.

Les rapports sont stockés dans le répertoire que vous avez choisi comme Emplacement dans la boîte de dialogue Nouvelle évaluation multiserveur.

Pour accéder aux rapports détaillés, vous pouvez parcourir les sous-répertoires, qui sont organisés par base de données source, nom de schéma et moteur de base de données cible.

Les rapports agrégés présentent des informations sur quatre colonnes concernant la complexité de conversion d'une base de données cible. Les colonnes incluent des informations sur la conversion d'objets de code, d'objets de stockage, d'éléments de syntaxe et de complexité de conversion.

L'exemple suivant montre les informations relatives à la conversion de deux schémas de base de données Oracle en Amazon RDS for PostgreSQL.

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

Les quatre mêmes colonnes sont ajoutées aux rapports pour chaque moteur de base de données cible supplémentaire spécifié.

Pour plus de détails sur la façon de lire ces informations, reportez-vous à la section suivante.

## Résultat pour un rapport d'évaluation agrégé

Le rapport agrégé d'évaluation de la migration des bases de données multiserveurs AWS Schema Conversion Tool est un fichier CSV contenant les colonnes suivantes :

- Server IP address and port
- Secret Manager key
- Name
- Description
- Database name
- Schema name
- Code object conversion % for *target\_database*
- Storage object conversion % for *target\_database*
- Syntax elements conversion % for *target\_database*
- Conversion complexity for *target\_database*

Pour recueillir des informations, AWS SCT exécute des rapports d'évaluation complets, puis agrège les rapports par schémas.

Dans le rapport, les trois champs suivants indiquent le pourcentage de conversion automatique possible sur la base de l'évaluation :

### % de conversion d'objets de code

Pourcentage d'objets de code du schéma qui AWS SCT peuvent être convertis automatiquement ou avec un minimum de modifications. Les objets de code incluent des procédures, des fonctions, des vues, etc.

### % de conversion des objets de stockage

Pourcentage d'objets de stockage que le SCT peut convertir automatiquement ou avec un minimum de modifications. Les objets de stockage incluent les tables, les index, les contraintes, etc.

### % de conversion des éléments de syntaxe

Pourcentage d'éléments syntaxiques que le SCT peut convertir automatiquement. Les éléments de syntaxe incluent SELECT FROMDELETE,,, et les JOIN clauses, etc.

Le calcul de la complexité de conversion est basé sur la notion d'éléments d'action. Un élément d'action reflète un type de problème détecté dans le code source que vous devez résoudre manuellement lors de la migration vers une cible particulière. Un élément d'action peut avoir plusieurs occurrences.

Une échelle pondérée identifie le niveau de complexité associé à la réalisation d'une migration. Le chiffre 1 représente le niveau de complexité le plus faible et le chiffre 10 représente le niveau de complexité le plus élevé.

# Conversion de schémas de base de données dans AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir vos schémas de base de données existants d'un moteur de base de données à un autre. La conversion d'une base de données à l'aide de AWS SCT l'interface utilisateur peut être assez simple, mais vous devez tenir compte de plusieurs éléments avant de procéder à la conversion.

Par exemple, vous pouvez AWS SCT effectuer les opérations suivantes :

- Vous pouvez l'utiliser AWS SCT pour copier un schéma de base de données sur site existant vers une instance de base de données Amazon RDS exécutant le même moteur. Cette fonction vous permet d'analyser les économies de coûts potentielles en cas de déplacement vers le cloud et de changement de votre type de licence.
- Dans certains cas, les fonctionnalités de base de données ne peuvent pas être converties en fonctionnalités Amazon RDS équivalentes. Si vous hébergez et gérez vous-même une base de données sur la plateforme Amazon Elastic Compute Cloud (Amazon EC2), vous pouvez émuler ces fonctionnalités en les remplaçant par des AWS services.
- AWS SCT automatise une grande partie du processus de conversion de votre schéma de base de données de traitement des transactions en ligne (OLTP) en une instance de base de données MySQL Amazon Relational Database Service (Amazon RDS), un cluster de base de données Amazon Aurora ou une instance de base de données PostgreSQL. Les moteurs de base de données source et cible contiennent de nombreuses fonctionnalités et capacités différentes et AWS SCT tentent de créer un schéma équivalent dans votre instance de base de données Amazon RDS dans la mesure du possible. Si aucune conversion directe n'est possible, AWS SCT fournit une liste des actions que vous pouvez entreprendre.

## Rubriques

- [Appliquer les règles de migration dans AWS Schema Conversion Tool](#)
- [Conversion de schémas à l'aide d'AWS SCT](#)
- [Conversion manuelle de schémas dans AWS SCT](#)
- [Mise à jour et actualisation des schémas convertis dans AWS SCT](#)
- [Enregistrer et appliquer des schémas convertis dans AWS SCT](#)
- [Comparaison de schémas dans AWS Schema Conversion Tool](#)



- [Affichage des objets transformés associés dans AWS Schema Conversion Tool](#)

AWS SCT prend en charge les conversions de traitement des transactions en ligne (OLTP) suivantes.

Base de données source	Base de données cible
IBM DB2 pour z/OS (version 12)	Édition compatible avec Amazon Aurora MySQL, édition compatible avec Amazon Aurora PostgreSQL, MySQL, PostgreSQL
IBM Db2 LUW (versions 9.1, 9.5, 9.7, 10.5, 11.1 et 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Base de données Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (version 2008 R2 et supérieure)	Aurora MySQL, Aurora PostgreSQL, Babelfish pour Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (version 5.5 et supérieure)	Aurora PostgreSQL, MySQL, PostgreSQL  Vous pouvez migrer le schéma et les données de MySQL vers un cluster de base de données Aurora MySQL sans utiliser AWS SCT. Pour plus d'informations, consultez la section <a href="#">Migration des données vers un cluster de base de données Amazon Aurora</a> .
Oracle (version 10.2 et supérieure)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (version 9.1 et supérieure)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12,5, 15,0, 15,5, 15,7 et 16,0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Pour plus d'informations sur la conversion d'un schéma d'entrepôt de données, consultez [Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT](#).

Pour convertir le schéma de votre base de données vers Amazon RDS, vous devez suivre les étapes de haut niveau suivantes :

- [Création de règles de migration dans AWS SCT](#)— Avant de convertir votre schéma AWS SCT, vous pouvez définir des règles qui modifient le type de données des colonnes, déplacent des objets d'un schéma à un autre et modifient le nom des objets.
- [???](#)— AWS SCT crée une version locale du schéma converti que vous pouvez consulter, mais il ne l'applique pas à votre instance de base de données cible tant que vous n'êtes pas prêt.
- [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#)— AWS SCT crée un rapport d'évaluation de la migration de base de données qui détaille les éléments du schéma qui ne peuvent pas être convertis automatiquement. Vous pouvez utiliser ce rapport pour identifier l'endroit où vous devez créer un schéma dans votre instance de base de données Amazon RDS compatible avec votre base de données source.
- [Conversion de schémas à l'aide d'AWS SCT](#)— Si vous avez des éléments de schéma qui ne peuvent pas être convertis automatiquement, vous avez deux choix : mettre à jour le schéma source puis reconverter, ou créer des éléments de schéma équivalents dans votre instance de base de données Amazon RDS cible.
- [Mise à jour et actualisation des schémas convertis dans AWS SCT](#)— Vous pouvez mettre à jour votre AWS SCT projet avec le schéma le plus récent de votre base de données source.
- [Enregistrer et appliquer des schémas convertis dans AWS SCT](#)— Lorsque vous êtes prêt, AWS SCT applique le schéma converti dans votre projet local à votre instance de base de données Amazon RDS cible.

## Appliquer les règles de migration dans AWS Schema Conversion Tool

Avant de convertir votre schéma avec AWS SCT, vous pouvez définir des règles de migration. Les règles de migration AWS SCT peuvent effectuer des transformations telles que modifier le type de données des colonnes, déplacer des objets d'un schéma à un autre et modifier le nom des objets. Supposons par exemple que vous disposiez d'un ensemble de tables dans votre schéma source nommé `test_TABLE_NAME`. Vous pouvez configurer une règle qui remplace le préfixe `test_` par le préfixe `demo_` dans le schéma cible.

**Note**

Vous ne pouvez créer des règles de migration que pour différents moteurs de base de données source et cible.

Vous pouvez créer des règles de migration qui exécutent les tâches suivantes :

- Ajouter, supprimer ou remplacer un préfixe
- Ajouter, supprimer ou remplacer un suffixe
- Modifier le classement des colonnes
- Modifier le type de données
- Modifier la longueur de `char`, `varchar`, et les types de `string` données
- Déplacer des objets
- Renommer des objets

Vous pouvez créer des règles de migration pour les objets suivants :

- Database (Base de données)
- Schema (Schéma)
- Tableau
- Colonne

## Création de règles de migration

Vous pouvez créer des règles de migration et les enregistrer dans le cadre de votre projet. Votre projet étant ouvert, utilisez la procédure suivante pour créer des règles de migration.

Pour créer des règles de migration

1. Dans le menu Affichage, choisissez Affichage cartographique.
2. Dans Mappages de serveurs, choisissez une paire de serveurs source et cible.
3. Choisissez Nouvelle règle de migration. La boîte de dialogue Règles de transformation apparaît.
4. Choisissez Add new rule. Une nouvelle ligne est ajoutée à la liste de règles.
5. Configurez votre règle :

- a. Pour Nom, saisissez un nom pour votre règle.
- b. Sous For, choisissez le type d'objet auquel la règle s'applique.
- c. Pour où, entrez un filtre à appliquer aux objets avant d'appliquer la règle de migration. La clause Where est évaluée à l'aide d'une clause Like. Vous pouvez entrer un nom exact pour sélectionner un objet, ou spécifier un modèle afin de sélectionner plusieurs objets.

Les champs disponibles pour la clause where varient en fonction du type de l'objet. Par exemple, s'il s'agit du type d'objet schéma, un seul champ est disponible pour le nom de schéma.

- d. Pour Actions, choisissez le type de règle de migration que vous souhaitez créer.
- e. En fonction du type de règle, entrez une ou deux valeurs supplémentaires. Par exemple, pour renommer un objet, entrez le nouveau nom de l'objet. Pour remplacer un préfixe, entrez l'ancien préfixe et le nouveau préfixe.

Pour les types de données char, varchar, nvarchar et string, vous pouvez modifier la longueur du type de données à l'aide de l'opérateur de multiplication. Par exemple, la `%*4` valeur transforme le type de `varchar(10)` données en `varchar(40)`.

6. Après avoir configuré votre règle de migration, choisissez Enregistrer pour enregistrer votre règle. Vous pouvez également choisir Cancel pour annuler vos modifications.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS [doesn't support](#) more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

**Transformation rule:** For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test\_%' add prefix 'demo\_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test\_%

Actions: add prefix demo\_%

Buttons: Save, Cancel, + Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Une fois que vous avez ajouté, modifié ou supprimé les règles, choisissez Save All pour enregistrer toutes vos modifications.
8. Choisissez Fermer pour fermer la boîte de dialogue des règles de transformation.

Vous pouvez utiliser l'icône à bascule pour désactiver une règle de migration sans la supprimer. Vous pouvez utiliser l'icône de copie pour dupliquer une règle de migration existante. Vous pouvez utiliser l'icône en forme de crayon pour modifier une règle de migration existante. Vous pouvez utiliser l'icône de suppression pour supprimer une règle de migration existante. Pour enregistrer les modifications que vous apportez à vos règles de migration, choisissez Enregistrer tout.

## Exportation des règles de migration

Si vous avez l'habitude de migrer vos données de votre base de données source vers votre base de données cible, vous pouvez fournir des informations sur vos règles de migration à AWS DMS. Pour plus d'informations sur les tâches, consultez la section [Utilisation des tâches de AWS Database Migration Service réplication](#).

Pour exporter les règles de migration

1. Dans le AWS Schema Conversion Tool, choisissez Mapping View dans le menu View.
2. Dans Règles de migration, choisissez une règle de migration, puis sélectionnez Modifier la règle de migration.
3. Choisissez Exporter le script pour AWS DMS.
4. Accédez à l'emplacement où vous souhaitez enregistrer votre script, puis choisissez Save. Vos règles de migration sont enregistrées sous forme de script JSON qui peut être utilisé par AWS DMS.

## Conversion de schémas à l'aide d'AWS SCT

Une fois que vous avez connecté votre projet à la fois à votre base de données source et à votre instance de base de données Amazon RDS cible, votre AWS Schema Conversion Tool projet affiche le schéma de votre base de données source dans le panneau de gauche. Le schéma est présenté sous forme d'arborescence, et chaque nœud de l'arborescence est « avec chargement différé ». Lorsque vous choisissez un nœud dans l'arborescence, il AWS SCT demande les informations de schéma à votre base de données source à ce moment-là.

Vous pouvez choisir des éléments de schéma de votre base de données source, puis convertir le schéma en schéma équivalent pour le moteur de base de données de votre instance DB cible. Vous pouvez choisir n'importe quel élément de schéma de votre base de données source de convertir. Si l'élément de schéma que vous choisissez dépend d'un élément parent, il génère AWS SCT également le schéma de l'élément parent. Supposons, par exemple, que vous choisissiez une table à convertir. Si tel est le cas, AWS SCT génère le schéma de la table et la base de données dans laquelle se trouve la table.

## Conversion de schémas

Pour convertir un schéma à partir de votre base de données source, cochez la case correspondant au nom du schéma à convertir. Choisissez ensuite ce schéma dans le panneau de gauche de votre projet. AWS SCT met en évidence le nom du schéma en bleu. Ouvrez le menu contextuel (clic droit) du schéma, puis choisissez Convertir le schéma, comme illustré ci-dessous.

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
  - Databases [12]
    - AdventureWorks2012\_CS
    - alfresco
    - GOLD\_TEST\_SS\_PG
    - LARGE\_DB\_SS
    - master
    - model
    - msdb
    - tempdb
    - TEST**
      - vmap
      - vpas
      - vrecon
    - Server Objects
    - SQL Server Agent
    - Applications
    - SQL Scripts
  - noSQL Clusters
  - ETL

Create mapping...  
 Create report  
**Convert schema**  
 Register agent  
 Compare schema  
 Load schema  
 Hide schema  
 Refresh from database  
 Collect statistics  
 Upload statistics  
 Create DMS task  
 Create Local & DMS task  
 Create Local task  
 Add virtual partitioning  
 Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

Une fois que vous avez converti le schéma de votre base de données source, vous pouvez choisir des éléments de schéma dans le volet gauche de votre projet et afficher le schéma converti dans les panneaux centraux de votre projet. Le panneau situé dans la partie centrale inférieure affiche les propriétés et la commande SQL pour créer le schéma converti, comme illustré ci-après.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the server structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > LARGE\_DB\_SS > Schemas [2] > dbo > Tables [4] > Account. The main pane shows the SQL script for the 'Account' table in the source database:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the script, it indicates 'Cursor position: 0 Modified: true'. The bottom pane shows the target Amazon RDS schema for the MySQL table 'Account':

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Une fois que vous avez converti votre schéma, vous pouvez enregistrer votre projet. Les informations de schéma de votre base de données source sont enregistrées avec votre projet. Cette fonctionnalité signifie que vous pouvez travailler hors ligne sans être connecté à votre base de données source. AWS SCT se connecte à votre base de données source pour mettre à jour le schéma dans votre projet si vous choisissez Refresh from Database (Actualiser depuis la base de données) pour votre base de données source. Pour de plus amples informations, veuillez consulter [Mise à jour et actualisation des schémas convertis dans AWS SCT](#).

Vous pouvez créer un rapport d'évaluation de la migration de base de données des éléments qui ne peuvent pas être convertis automatiquement. Le rapport d'évaluation est utile pour identifier et résoudre les éléments de schéma qui ne peut pas être convertis automatiquement. Pour de plus amples informations, veuillez consulter [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#).

Lorsqu'il AWS SCT génère un schéma converti, il ne l'applique pas immédiatement à l'instance de base de données cible. En revanche, le schéma converti est stocké localement jusqu'à ce que vous soyez prêt à l'appliquer à l'instance DB cible. Pour de plus amples informations, veuillez consulter [Appliquer votre schéma converti](#).

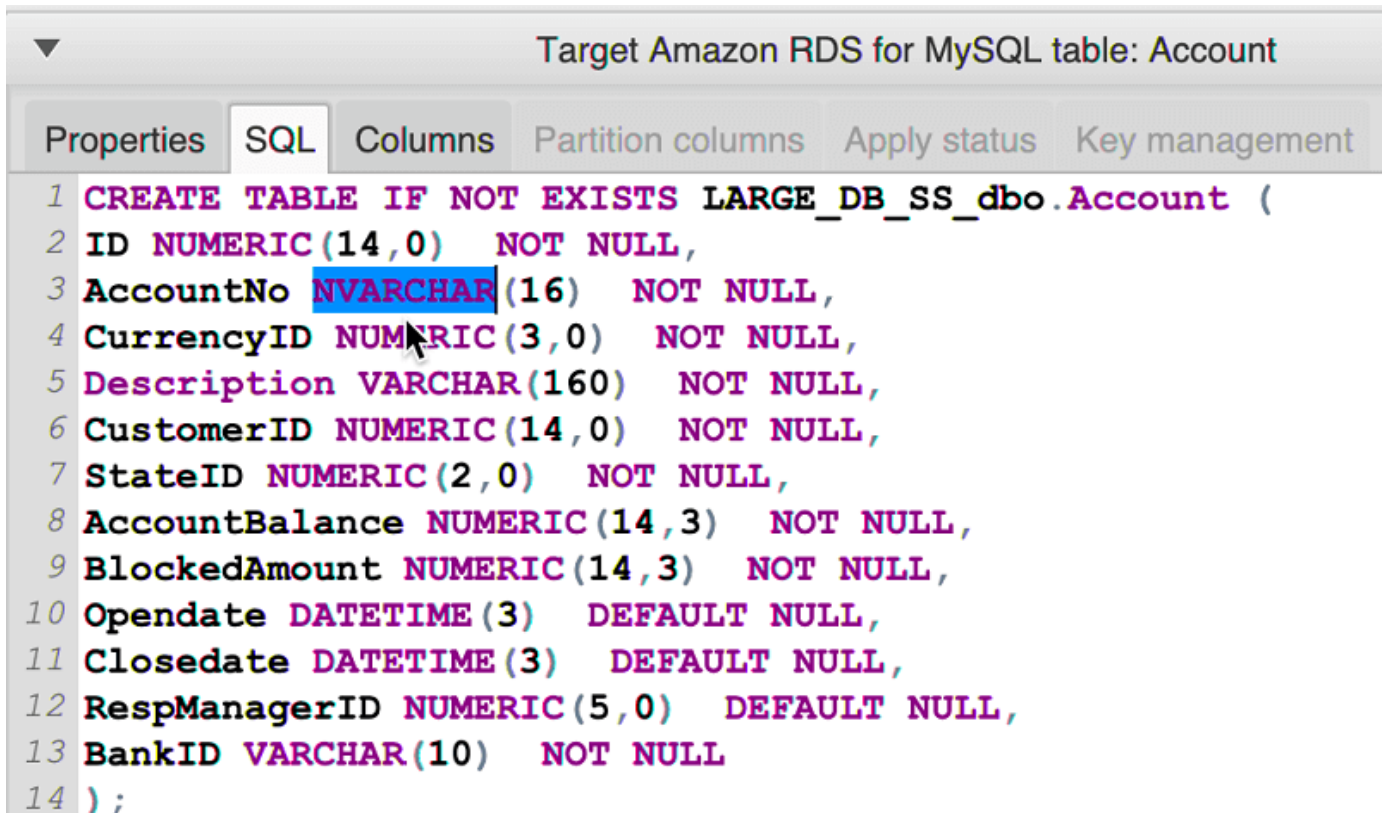


## Modification du schéma converti

Vous pouvez modifier le schéma converti et enregistrer les modifications dans le cadre de votre projet.

Pour modifier le schéma converti

1. Dans le volet gauche qui affiche le schéma de votre base de données source, sélectionnez l'élément pour lequel vous voulez modifier le schéma converti.
2. Dans le volet situé dans la partie centrale inférieure qui affiche le schéma converti pour l'élément sélectionné, choisissez l'onglet SQL.
3. Dans le texte affiché pour l'onglet SQL, modifiez le schéma comme requis. Le schéma est automatiquement enregistré avec votre projet comme vous le mettez à jour.



The screenshot shows the AWS Schema Conversion Tool interface. At the top, it says "Target Amazon RDS for MySQL table: Account". Below this, there are several tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". The "SQL" tab is selected. The SQL text displayed is:

```
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (  
2 ID NUMERIC(14,0) NOT NULL,  
3 AccountNo NVARCHAR(16) NOT NULL,  
4 CurrencyID NUMERIC(3,0) NOT NULL,  
5 Description VARCHAR(160) NOT NULL,  
6 CustomerID NUMERIC(14,0) NOT NULL,  
7 StateID NUMERIC(2,0) NOT NULL,  
8 AccountBalance NUMERIC(14,3) NOT NULL,  
9 BlockedAmount NUMERIC(14,3) NOT NULL,  
10 Opendate DATETIME(3) DEFAULT NULL,  
11 Closedate DATETIME(3) DEFAULT NULL,  
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,  
13 BankID VARCHAR(10) NOT NULL  
14 );
```

Les modifications que vous apportez au schéma converti sont stockées avec votre projet lorsque vous effectuez des mises à jour. Si vous convertissez nouvellement un élément de schéma de votre base de données source et que vous avez fait des mises à jour de schéma précédemment converti pour cet élément, ces mises à jour existantes sont remplacées par l'élément de schéma nouvellement converti basé sur votre base de données source.

## Effacement d'un schéma converti

Jusqu'à ce que vous appliquiez le schéma à votre instance de base de données cible, le schéma converti AWS SCT ne sera stocké que localement dans votre projet. Vous pouvez effacer le schéma planifié de votre projet en choisissant le nœud arborescent pour votre instance de base de données, puis en choisissant Actualiser depuis la base de données. Comme aucun schéma n'a été écrit sur votre instance de base de données cible, l'actualisation à partir de la base de données supprime les éléments de schéma planifiés de votre AWS SCT projet pour qu'ils correspondent à ceux qui existent dans votre instance de base de données source.

## Conversion manuelle de schémas dans AWS SCT

Le rapport d'évaluation inclut une liste d'éléments qui ne peuvent pas être convertis automatiquement dans le moteur de base de données de votre instance de base de données Amazon RDS cible. Pour chaque élément qui ne peut pas être converti, il y a un élément de l'action sur l'onglet Action Items.

Vous pouvez réagir aux éléments d'action du rapport d'évaluation des manières suivantes :

- Modifiez votre schéma de base de données source.
- Modifiez votre schéma de base de données cible.

## Modification de votre schéma source

Pour certains éléments, il peut être plus facile de modifier le schéma de base de données de votre base de données source en un schéma pouvant être converti automatiquement. Tout d'abord, vérifiez que les nouvelles modifications sont compatibles avec l'architecture de votre application, puis mettez à jour le schéma dans votre base de données source. Enfin, actualisez votre projet avec les informations de schéma mises à jour. Vous pouvez ensuite convertir le schéma mis à jour et générer un nouveau rapport d'évaluation de la migration de la base de données. Les éléments d'action n'apparaissent plus pour les éléments qui ont été modifiés dans le schéma source.

L'avantage de ce processus est que le schéma mis à jour est toujours disponible lors de l'actualisation à partir de votre base de données source.

## Modification de votre schéma cible

Pour certains éléments, il peut être plus facile d'appliquer le schéma converti à votre base de données cible, et d'ajouter ensuite les éléments de schéma équivalent manuellement à votre base

de données cible pour les éléments qui n'ont pas pu être convertis automatiquement. Vous pouvez écrire tout le schéma qui peut être converti automatiquement à votre instance DB cible en appliquant le schéma. Pour de plus amples informations, veuillez consulter [Enregistrer et appliquer des schémas convertis dans AWS SCT](#).

Le schéma qui est écrit dans votre instance DB cible ne contient pas les éléments qui ne peuvent pas être convertis automatiquement. Après avoir appliqué le schéma à votre instance DB cible, vous pouvez alors créer manuellement un schéma dans votre instance DB cible qui est équivalent à celui de la base de données source. Les éléments d'action dans le rapport d'évaluation de la migration de base de données contiennent des suggestions sur la façon de créer le schéma équivalent.

#### Warning

Si vous créez manuellement le schéma de votre instance DB cible, enregistrez une copie de n'importe quel travail manuel que vous faites. Si vous appliquez de nouveau le schéma converti à partir de votre projet à votre instance DB cible, il remplace le travail manuel que vous avez terminé.

Dans certains cas, vous ne pouvez pas créer de schéma équivalent dans votre instance DB cible. Vous devrez reconcevoir une partie de votre application et de votre base de données afin d'utiliser les fonctionnalités disponibles dans le moteur de base de données pour votre instance de base de données cible. Dans d'autres cas, vous pourrez simplement ignorer le schéma qui ne peut pas être converti automatiquement.

## Mise à jour et actualisation des schémas convertis dans AWS SCT

Vous pouvez mettre à jour le schéma source et le schéma cible de votre AWS Schema Conversion Tool projet.

- **Source** : si vous mettez à jour le schéma de votre base de données source, AWS SCT remplace le schéma de votre projet par le schéma le plus récent de votre base de données source. A l'aide de cette fonctionnalité, vous pouvez mettre à jour votre projet si les modifications ont été apportées au schéma de votre base de données source.
- **Cible** : si vous mettez à jour le schéma de votre instance de base de données Amazon RDS cible AWS SCT, remplacez le schéma de votre projet par le dernier schéma de votre instance de base de données cible. Si vous n'avez appliqué aucun schéma à votre instance de base de

données cible, AWS SCT efface le schéma converti de votre projet. Vous pouvez ensuite convertir le schéma de votre base de données source en une instance DB cible propre.

Vous pouvez mettre à jour le schéma de votre AWS SCT projet en choisissant Actualiser depuis la base de données.

#### Note

Lorsque vous actualisez votre schéma, AWS SCT charge les métadonnées uniquement lorsque cela est nécessaire. Pour charger complètement l'ensemble du schéma de votre base de données, ouvrez le menu contextuel (clic droit) correspondant à votre schéma, puis choisissez Charger le schéma. Par exemple, vous pouvez utiliser cette option pour charger les métadonnées de votre base de données en une seule fois, puis travailler hors connexion.

## Enregistrer et appliquer des schémas convertis dans AWS SCT

Lorsque le schéma converti est AWS Schema Conversion Tool généré (comme indiqué dans [???](#)), il n'applique pas immédiatement le schéma converti à l'instance de base de données cible. Au lieu de cela, le schéma converti est stocké localement dans votre projet jusqu'à ce que vous soyez prêt à l'appliquer à l'instance DB cible. A l'aide de cette fonctionnalité, vous pouvez travailler avec des éléments de schéma qui ne peuvent pas être convertis automatiquement à votre moteur de base de données cible. Pour plus d'informations sur les éléments qui ne peuvent pas être convertis automatiquement, consultez [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#).

De manière facultative, l'outil peut enregistrer votre schéma converti en un fichier en tant que script SQL avant d'appliquer le schéma à votre instance DB cible. Vous pouvez faire appliquer par l'outil le schéma converti directement à votre instance DB cible.

### Enregistrement de votre schéma converti dans un fichier

Vous pouvez enregistrer votre schéma converti sous forme de scripts SQL dans un fichier texte. En utilisant cette approche, vous pouvez modifier les scripts SQL générés AWS SCT pour traiter les éléments que l'outil ne peut pas convertir automatiquement. Vous pouvez ensuite exécuter vos scripts mis à jour sur votre instance DB cible pour appliquer votre schéma converti à votre base de données cible.

## Pour enregistrer votre schéma converti sous forme de scripts SQL

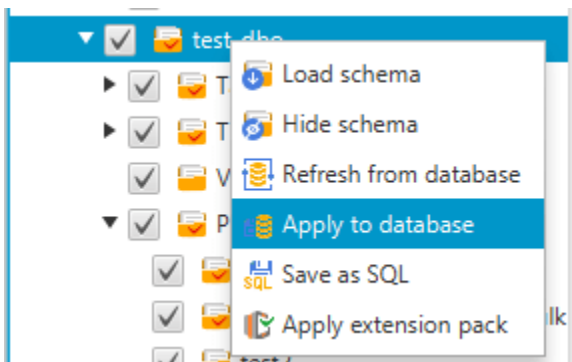
1. Choisissez votre schéma et ouvrez le menu contextuel (clic droit).
2. Choisissez Enregistrer en tant que SQL.
3. Entrez le nom du fichier et choisissez Enregistrer.
4. Enregistrez votre schéma converti à l'aide de l'une des options suivantes :
  - Fichier unique
  - Un seul fichier par étape
  - Un seul fichier par relevé

## Pour choisir le format du script SQL

1. Dans le menu Paramètres, choisissez Paramètres du projet.
2. Choisissez Enregistrer les scripts.
3. Pour Vendor, choisissez la plate-forme de base de données.
4. Pour Enregistrer les scripts SQL sur, choisissez la manière dont vous souhaitez enregistrer le script de schéma de base de données.
5. Cliquez sur OK pour enregistrer les paramètres.

## Appliquer votre schéma converti

Lorsque vous êtes prêt à appliquer votre schéma converti à votre instance de base de données Amazon RDS cible, choisissez l'élément de schéma dans le panneau de droite de votre projet. Ouvrez le menu contextuel (clic droit) pour l'élément de schéma, puis choisissez Apply to database, comme indiqué ci-dessous.



## Le schéma du pack d'extension

La première fois que vous appliquez votre schéma converti à votre instance de base de données cible, AWS SCT ajoute un schéma supplémentaire à cette dernière. Ce schéma met en œuvre les fonctions système de la base de données source nécessaires lors de l'écriture du schéma converti dans l'instance DB cible. Le schéma est appelé schéma de kit d'extension.

Ne modifiez pas ce schéma ou vous risquez d'obtenir des résultats inattendus dans le schéma converti écrit dans l'instance de base de données cible. Lorsque votre schéma est entièrement migré vers votre instance de base de données cible et que vous n'en avez plus besoin AWS SCT, vous pouvez supprimer le schéma du pack d'extension.

Le schéma de kit d'extension est nommé selon votre base de données source, comme suit :

- IBM DB2 LUW : `aws_db2_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- ÉTUI À SAVON : `aws_sapase_ext`

Pour de plus amples informations, veuillez consulter [Utilisation des AWS Lambda fonctions du pack d' AWS SCT extension](#) .

## Comparaison de schémas dans AWS Schema Conversion Tool

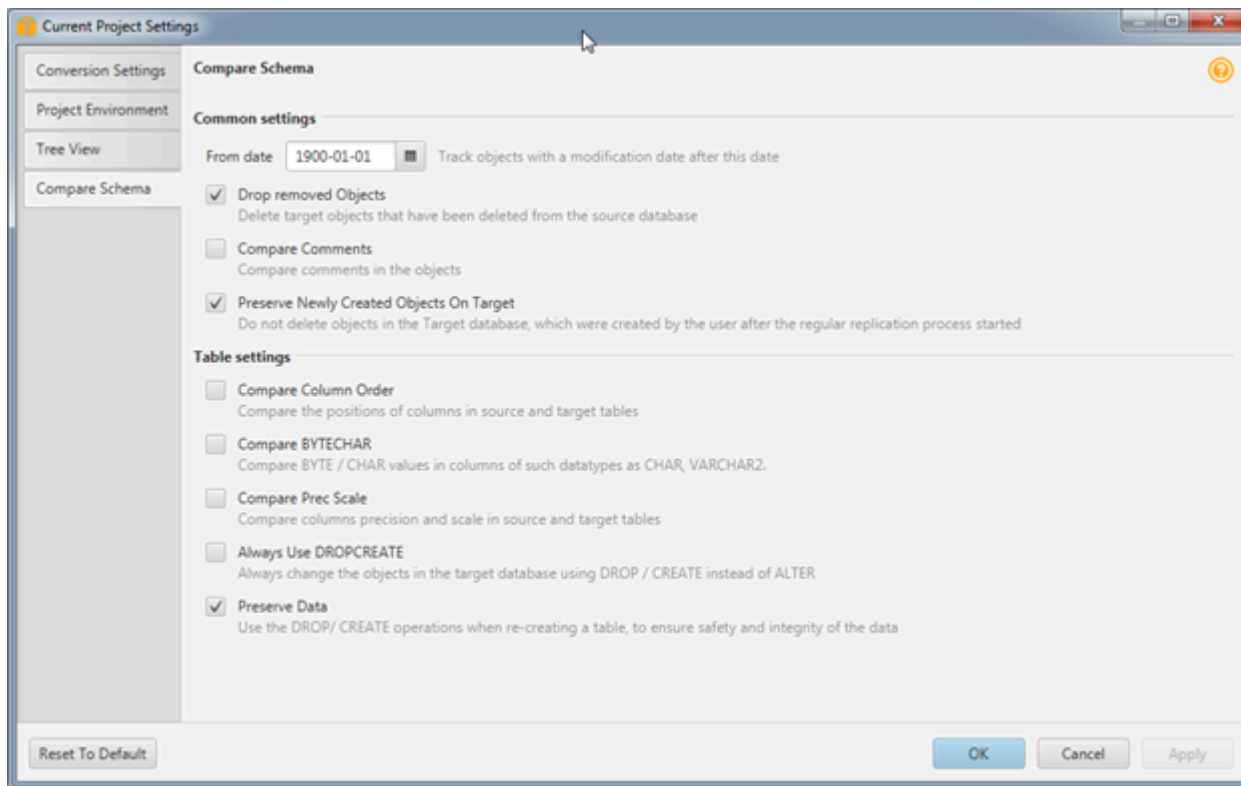
Si vous avez apporté des modifications à votre schéma source ou cible après la migration, vous pouvez comparer les deux schémas de base de données à l'aide de. AWS SCT Vous pouvez comparer les schémas de versions identiques ou antérieures au schéma source.

Les comparaisons de schémas suivantes sont prises en charge :

- Oracle avec Oracle, versions 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0, 10
- SQL Server vers SQL Server, versions 2016, 2014, 2012, 2008 RD2, 2008
- Édition compatible avec PostgreSQL vers PostgreSQL et Aurora PostgreSQL, versions 9.6, 9.5.9, 9.5.4

- MySQL avec MySQL, versions 5.6.36, 5.7.17, 5.5

Vous pouvez spécifier des paramètres pour la comparaison de schémas sur l'onglet Compare Schema (Comparer un schéma) de la page Project Settings (Paramètres du projet).

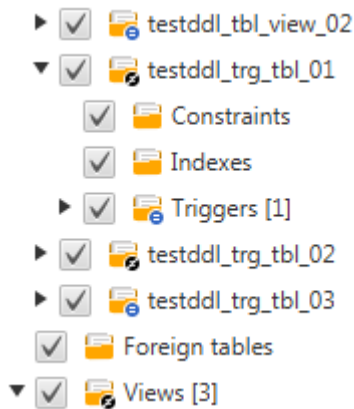


Pour comparer des schémas, vous sélectionnez les schémas et vous indiquez les AWS SCT objets qui diffèrent entre les deux schémas et les objets qui ne le sont pas.

Pour comparer deux schémas

1. Ouvrez un projet AWS SCT existant ou créez un projet et connectez-vous aux points de terminaison source et cible.
2. Choisissez le schéma que vous voulez comparer.
3. Ouvrez le menu contextuel (clic droit) et choisissez Comparer le schéma.

AWS SCT indique les objets différents entre les deux schémas en ajoutant un cercle noir à l'icône de l'objet.



Vous pouvez appliquer les résultats de la comparaison de schémas à un seul objet, à une seule catégorie d'objets, ou à l'ensemble du schéma. Choisissez la case en regard de la catégorie, de l'objet ou du schéma auquel vous voulez appliquer les résultats.

## Affichage des objets transformés associés dans AWS Schema Conversion Tool

Après une conversion de schéma, dans certains cas, plusieurs objets AWS SCT peuvent avoir été créés pour un objet de schéma dans la base de données source. Par exemple, lorsque vous effectuez une conversion d'Oracle vers PostgreSQL AWS SCT, chaque déclencheur Oracle est transformé en déclencheur et en fonction de déclenchement sur la cible PostgreSQL. En outre, lors de la AWS SCT conversion d'une fonction ou d'une procédure de package Oracle en PostgreSQL, il crée une fonction équivalente et une fonction INIT qui doivent être exécutées en tant que bloc d'initialisation avant que la procédure ou la fonction ne puisse être exécutée.

La procédure suivante vous permet de voir tous les objets associés qui ont été créés après une conversion de schéma.

Pour afficher les objets associés qui ont été créés lors d'une conversion de schémas

1. Après la conversion de schémas, choisissez l'objet converti dans l'arborescence cible.
2. Choisissez l'onglet Related Converted Objects (Objets convertis associés).
3. Affichez la liste des objets cible associés.



# Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT

Le AWS Schema Conversion Tool (AWS SCT) automatise une grande partie du processus de conversion de votre schéma d'entrepôt de données en schéma de base de données Amazon RDS ou Aurora. Étant donné que les moteurs de base de données source et cible peuvent avoir de nombreuses fonctionnalités différentes, essayez AWS SCT de créer un schéma équivalent dans votre base de données cible dans la mesure du possible. Si aucune conversion directe n'est possible, AWS SCT fournit un rapport d'évaluation contenant une liste des mesures que vous pouvez prendre. À l'aide de AWS SCT, vous pouvez gérer les clés, cartographier les types de données et les objets, et créer des conversions manuelles.

AWS SCT peut convertir les schémas d'entrepôt de données suivants.

- Amazon Redshift
- Azure Synapse Analytics (version 10)
- BigQuery
- Base de données Greenplum (version 4.3)
- Microsoft SQL Server (version 2008 et supérieure)
- Netezza (version 7.0.3 et supérieure)
- Oracle (version 10.2 et supérieure)
- Flocon de neige (version 3)
- Teradata (version 13 et supérieure)
- Vertica (version 7.2 et supérieure)

Pour plus d'informations sur la conversion d'un schéma de base de données de traitement des transactions en ligne (OLTP), consultez [Conversion de schémas de base de données dans AWS Schema Conversion Tool](#).

Pour convertir un schéma d'entrepôt de données, procédez comme suit :

1. Spécifiez la stratégie et les règles d'optimisation, ainsi que les règles de migration que vous AWS SCT souhaitez utiliser. Vous pouvez définir des règles qui changent le type de données des colonnes, déplacent des objets d'un schéma vers un autre et modifient les noms des objets.

Vous pouvez définir des règles d'optimisation et de migration dans les paramètres. Pour plus d'informations sur les stratégies d'optimisation, voir [Choix des stratégies et des règles d'optimisation à utiliser avec AWS SCT](#). Pour plus d'informations sur les règles de migration, voir [Création de règles de migration dans AWS SCT](#)

2. Indiquez les statistiques relatives à votre entrepôt de données source afin qu' AWS SCT puisse optimiser la conversion de votre entrepôt de données. Vous pouvez collecter les statistiques directement à partir de la base de données, ou charger un fichier de statistiques existant. Pour plus d'informations sur l'indication des statistiques relatives à l'entrepôt de données, consultez [Collecte ou téléchargement de statistiques pour AWS SCT](#).
3. Créez un rapport d'évaluation de la migration de base de données qui détaille les éléments de schéma qui ne peuvent pas être convertis automatiquement. Vous pouvez utiliser ce rapport pour identifier où vous avez besoin de créer manuellement un schéma dans votre base de données cible qui est compatible avec votre base de données source. Pour plus d'informations sur le rapport d'évaluation, consultez [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#).
4. Convertir le schéma. AWS SCT crée une version locale du schéma converti pour vous permettre de le vérifier, mais il ne l'applique pas à votre base de données cible tant que vous n'êtes pas prêt. Pour plus d'informations sur la conversion, consultez [Conversion de votre schéma à l'aide de AWS SCT](#).
5. Une fois que vous convertissez votre schéma, vous pouvez gérer et modifier vos clés. La gestion des clés est le cœur d'une conversion d'entrepôt de données. Pour plus d'informations sur la gestion des clés, consultez [Gestion et personnalisation des clés dans AWS SCT](#).
6. Si vous avez des éléments de schéma qui ne peuvent pas être convertis automatiquement, deux choix s'offrent à vous : mettre à jour le schéma source, puis relancer la conversion, ou créer des éléments de schéma équivalents dans votre base de données cible. Pour plus d'informations sur la conversion manuelle des éléments de schéma, consultez [Gestion des conversions manuelles dans AWS SCT](#). Pour plus d'informations sur la mise à jour de votre schéma source, consultez [Mise à jour et actualisation de votre schéma converti dans AWS SCT](#).
7. Lorsque vous êtes prêt, vous pouvez appliquer le schéma converti à votre base de données cible. Pour plus d'informations sur l'enregistrement et l'application du schéma converti, consultez [Enregistrer et appliquer votre schéma converti dans AWS SCT](#).

## Autorisations pour Amazon Redshift en tant que cible

Les autorisations requises pour Amazon Redshift en tant que cible sont répertoriées ci-dessous :

- CREATE ON DATABASE — permet de créer de nouveaux schémas dans la base de données.
- CREATE ON SCHEMA — permet de créer des objets dans le schéma de base de données.
- UTILISATION DES SUBVENTIONS EN FONCTION DE LA LANGUE : permet de créer de nouvelles fonctions et procédures dans la base de données.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg\_catalog : fournit à l'utilisateur des informations système sur le cluster Amazon Redshift.
- GRANT SELECT ON pg\_class\_info — fournit à l'utilisateur des informations sur le style de distribution des tables.

Vous pouvez utiliser l'exemple de code suivant pour créer un utilisateur de base de données et accorder les autorisations.

```
CREATE USER user_name PASSWORD your_password;  
GRANT CREATE ON DATABASE db_name TO user_name;  
GRANT CREATE ON SCHEMA schema_name TO user_name;  
GRANT USAGE ON LANGUAGE plpythonu TO user_name;  
GRANT USAGE ON LANGUAGE plpgsql TO user_name;  
GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO user_name;  
GRANT SELECT ON pg_class_info TO user_name;  
GRANT SELECT ON sys_serverless_usage TO user_name;  
GRANT SELECT ON pg_database_info TO user_name;  
GRANT SELECT ON pg_statistic TO user_name;
```

Dans l'exemple précédent, remplacez *user\_name* par le nom de votre utilisateur. Remplacez-le ensuite *db\_name* par le nom de votre base de données Amazon Redshift cible. Remplacez-le ensuite *schema\_name* par le nom de votre schéma Amazon Redshift. Répétez l'opération GRANT CREATE ON SCHEMA pour chaque schéma cible dans lequel vous allez appliquer le code converti ou migrer des données. Enfin, remplacez-le *your\_password* par un mot de passe sécurisé.

Vous pouvez appliquer un pack d'extension à votre base de données Amazon Redshift cible. Le pack d'extension est un module complémentaire qui émule les fonctions de la base de données source requises pour convertir des objets vers la base de données cible. Pour de plus amples informations, veuillez consulter [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

Pour cette opération, vous AWS SCT devez avoir l'autorisation d'accéder au compartiment Amazon S3 en votre nom. Pour fournir cette autorisation, créez un utilisateur AWS Identity and Access Management (IAM) avec la politique suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-sct-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": ""
    }
  ]
}
```

## Choix des stratégies et des règles d'optimisation à utiliser avec AWS SCT

Pour optimiser la façon dont le schéma de votre entrepôt de données est AWS Schema Conversion Tool converti, vous pouvez choisir les stratégies et les règles que vous souhaitez que l'outil utilise. Après la conversion de votre schéma et la vérification des clés suggérés, vous pouvez ajuster vos règles ou modifier votre stratégie afin d'obtenir les résultats souhaités.

Pour choisir vos stratégies d'optimisation et règles

1. Choisissez Settings, puis Projet Settings. La boîte de dialogue Current project settings s'affiche.

2. Dans le volet gauche, choisissez Optimization Strategies. Les stratégies d'optimisation apparaissent dans le volet droit avec les valeurs par défaut sélectionnées.
3. Pour Strategy Sector, choisissez la stratégie d'optimisation que vous voulez utiliser. Sélectionnez parmi les éléments suivants :
  - Utiliser les métadonnées, ignorer les informations statistiques — Dans cette stratégie, seules les informations issues des métadonnées sont utilisées pour les décisions d'optimisation. Par exemple, s'il existe plusieurs index sur une table source, l'ordre de tri de la base de données source est utilisé, et le premier index devient une clé de distribution.
  - Ignorer les métadonnées, utiliser des informations statistiques — Dans cette stratégie, les décisions d'optimisation sont dérivées uniquement d'informations statistiques. Cette stratégie s'applique uniquement aux tables et aux colonnes pour lesquelles les statistiques sont fournis. Pour de plus amples informations, veuillez consulter [Collecte ou téléchargement de statistiques pour AWS SCT](#).
  - Utiliser des métadonnées et utiliser des informations statistiques — Dans cette stratégie, les métadonnées et les statistiques sont utilisées pour les décisions d'optimisation.
4. Une fois que vous choisissez votre stratégie d'optimisation, vous pouvez choisir les règles que vous voulez utiliser. Sélectionnez parmi les éléments suivants :
  - Choix de la clé de distribution et des clés de tri en utilisant les métadonnées
  - Choix de la table des faits et la dimension appropriée pour le classement
  - Analyse de la cardinalité des colonnes d'index
  - Trouvez les tables et les colonnes les plus utilisées dans la table du journal des requêtes

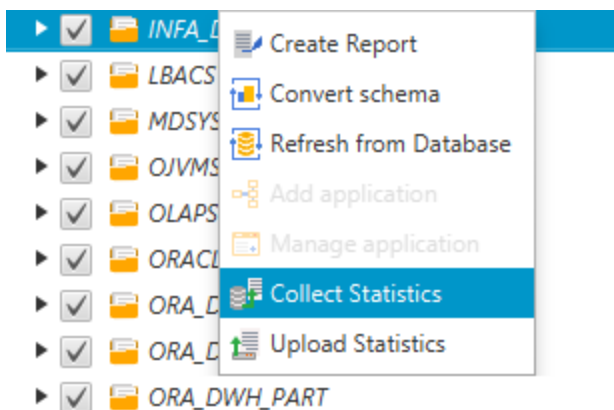
Pour chaque règle, vous pouvez saisir un poids pour la clé de tri et un poids pour la clé de distribution. AWS SCT utilise les poids choisis lors de la conversion de votre schéma. Plus tard, lorsque vous vérifiez les clés suggérées, vous pouvez revenir ici et modifier vos paramètres si vous n'êtes pas satisfait des résultats. Pour de plus amples informations, veuillez consulter [Gestion et personnalisation des clés dans AWS SCT](#).

# Collecte ou téléchargement de statistiques pour AWS SCT

Pour optimiser la façon dont le schéma de votre entrepôt de données est AWS Schema Conversion Tool converti, vous pouvez fournir des statistiques à partir de votre base de données source que l'outil peut utiliser. Vous pouvez collecter les statistiques directement à partir de la base de données, ou charger un fichier de statistiques existant.

Pour fournir et vérifier les statistiques

1. Ouvrez votre projet et connectez-vous à votre base de données source.
2. Choisissez un objet de schéma dans le volet gauche de votre projet et ouvrez le menu de contexte (clic droit) pour l'objet. Choisissez Collect Statistics ou Upload Statistics comme le suggère l'illustration ci-dessous.



3. Choisissez un objet de schéma dans le volet gauche de votre projet, puis choisissez l'onglet Statistics. Vous pouvez vérifier les statistiques pour l'objet.

Column Name	Stats Collection Date	Stats collection mode	Stats usage count	Stats cardinality
PART_ID	2016-06-14 15:41:23	online		9000
ADJUSTER_ID	2016-06-14 15:41:23	online		24
SPEC_ID	2016-06-14 15:41:23	online		111

Plus tard, lorsque vous vérifiez les clés suggérées, vous pouvez collecter des statistiques supplémentaires et répéter l'opération si vous n'êtes pas satisfait des résultats. Pour de plus amples informations, veuillez consulter [Gestion et personnalisation des clés dans AWS SCT](#).

## Création de règles de migration dans AWS SCT

Avant de convertir votre schéma avec AWS SCT, vous pouvez définir des règles de migration. Les règles de migration peuvent notamment modifier le type de données des colonnes, déplacer des objets d'un schéma à un autre et modifier le nom des objets. Supposons par exemple que vous disposiez d'un ensemble de tables dans votre schéma source nommé test\_TABLE\_NAME. Vous pouvez configurer une règle qui remplace le préfixe test\_ par le préfixe demo\_ dans le schéma cible.

### Note

Vous pouvez créer des règles de migration uniquement pour différents moteurs de base de données source et cible.

Vous pouvez créer des règles de migration qui exécutent les tâches suivantes :

- Ajouter, supprimer ou remplacer un préfixe
- Ajouter, supprimer ou remplacer un suffixe
- Modifier le classement des colonnes
- Modifier le type de données
- Modifier la longueur de `char`, `varchar`, et les types de `string` données
- Déplacer des objets
- Renommer des objets

Vous pouvez créer des règles de migration pour les objets suivants :

- Database (Base de données)
- Schema (Schéma)
- Tableau
- Colonne

## Création de règles de migration

Vous pouvez créer des règles de migration et les enregistrer dans le cadre de votre projet. Votre projet étant ouvert, utilisez la procédure suivante pour créer des règles de migration.

Pour créer des règles de migration

1. Dans le menu Affichage, choisissez Affichage cartographique.
2. Dans Mappages de serveurs, choisissez une paire de serveurs source et cible.
3. Choisissez Nouvelle règle de migration. La boîte de dialogue Règles de transformation apparaît.
4. Choisissez Add new rule. Une nouvelle ligne est ajoutée à la liste de règles.
5. Configurez votre règle :
  - a. Pour Nom, saisissez un nom pour votre règle.
  - b. Sous For, choisissez le type d'objet auquel la règle s'applique.
  - c. Pour où, entrez un filtre à appliquer aux objets avant d'appliquer la règle de migration. La clause Where est évaluée à l'aide d'une clause Like. Vous pouvez entrer un nom exact pour sélectionner un objet, ou spécifier un modèle afin de sélectionner plusieurs objets.



Les champs disponibles pour la clause where varient en fonction du type de l'objet. Par exemple, s'il s'agit du type d'objet schéma, un seul champ est disponible pour le nom de schéma.

- d. Pour Actions, choisissez le type de règle de migration que vous souhaitez créer.
  - e. En fonction du type de règle, entrez une ou deux valeurs supplémentaires. Par exemple, pour renommer un objet, entrez le nouveau nom de l'objet. Pour remplacer un préfixe, entrez l'ancien préfixe et le nouveau préfixe.
6. Après avoir configuré votre règle de migration, choisissez Enregistrer pour enregistrer votre règle. Vous pouvez également choisir Cancel pour annuler vos modifications.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS *doesn't support* more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

**Transformation rule:** For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test\_%' add prefix 'demo\_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test\_%

Actions: add prefix demo\_%

Buttons: Save, Cancel, + Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Une fois que vous avez ajouté, modifié ou supprimé les règles, choisissez Save All pour enregistrer toutes vos modifications.
8. Choisissez Fermer pour fermer la boîte de dialogue des règles de transformation.

Vous pouvez utiliser l'icône à bascule pour désactiver une règle de migration sans la supprimer. Vous pouvez utiliser l'icône de copie pour dupliquer une règle de migration existante. Vous pouvez utiliser l'icône en forme de crayon pour modifier une règle de migration existante. Vous pouvez utiliser l'icône de suppression pour supprimer une règle de migration existante. Pour enregistrer les modifications que vous apportez à vos règles de migration, choisissez Enregistrer tout.

## Exportation des règles de migration

Si vous utilisez AWS Database Migration Service (AWS DMS) pour migrer vos données de votre base de données source vers votre base de données cible, vous pouvez fournir des informations sur vos règles de migration à AWS DMS. Pour plus d'informations sur les tâches, consultez la section [Utilisation des tâches de AWS Database Migration Service réplication](#).

Pour exporter les règles de migration

1. Dans le AWS Schema Conversion Tool, choisissez Mapping View dans le menu View.
2. Dans Règles de migration, choisissez une règle de migration, puis sélectionnez Modifier la règle de migration.
3. Choisissez Exporter le script pour AWS DMS.
4. Accédez à l'emplacement où vous souhaitez enregistrer votre script, puis choisissez Save. Vos règles de migration sont enregistrées sous forme de script JSON qui peut être utilisé par AWS DMS.

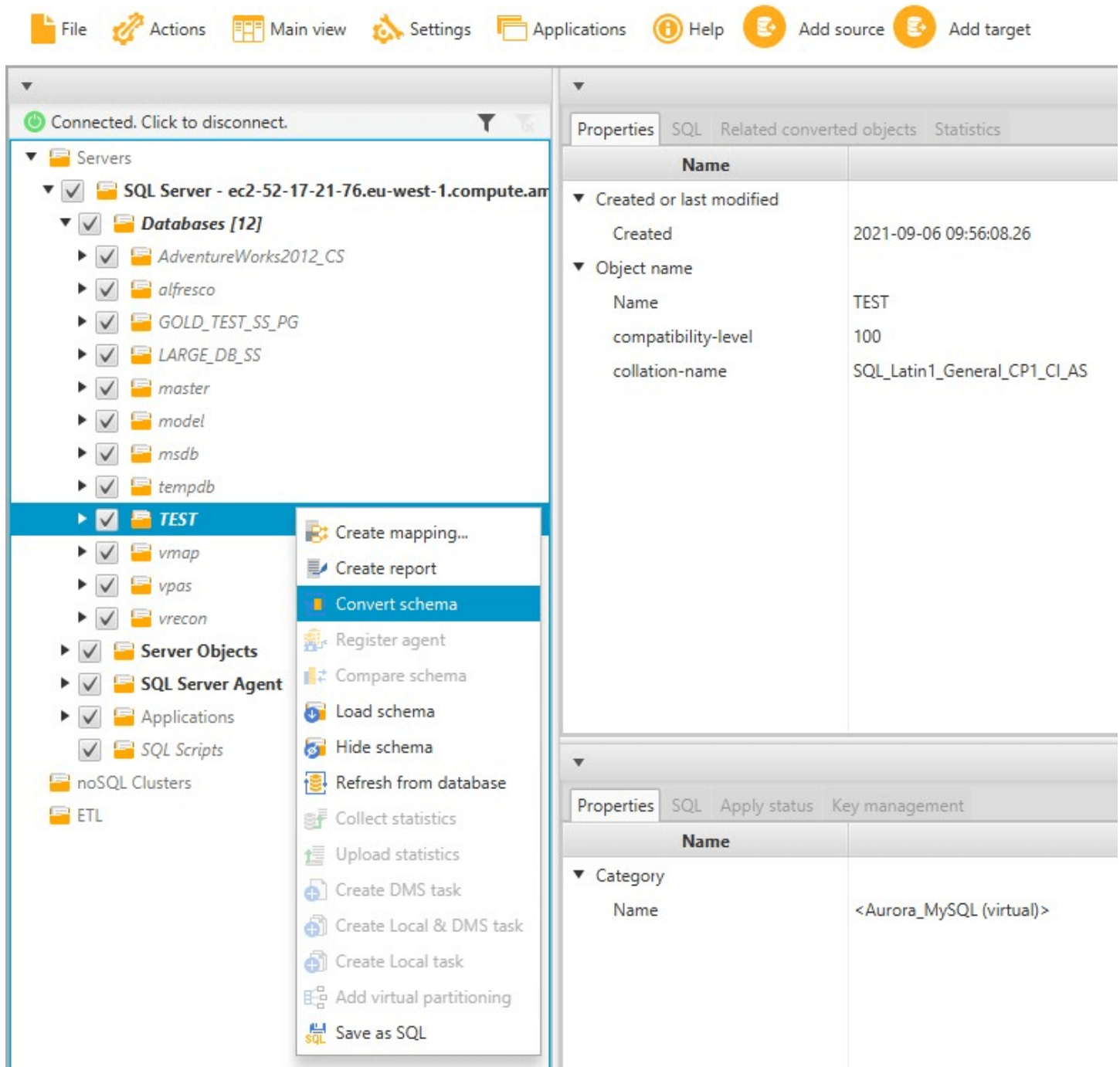
## Conversion de votre schéma à l'aide de AWS SCT

Après avoir connecté votre projet à la fois à votre base de données source et à votre base de données cible, votre AWS Schema Conversion Tool projet affiche le schéma de votre base de données source dans le panneau de gauche. Le schéma est présenté sous forme d'arborescence, et chaque nœud de l'arborescence est « avec chargement différé ». Lorsque vous sélectionnez un nœud dans l'arborescence, AWS SCT demande les informations de schéma de votre base de données source à ce moment-là.

Vous pouvez choisir des éléments de schéma de votre base de données source, puis convertir le schéma en schéma équivalent pour le moteur de base de données de votre base de données cible. Vous pouvez choisir n'importe quel élément de schéma de votre base de données source de convertir. Si l'élément de schéma que vous choisissez dépend d'un élément parent, il génère AWS SCT également le schéma de l'élément parent. Par exemple, si vous choisissez une colonne dans une table à convertir, vous AWS SCT générez le schéma de la colonne, la table dans laquelle se trouve la colonne et la base de données dans laquelle se trouve la table.

## Conversion de schémas

Pour convertir un schéma à partir de votre base de données source, cochez la case correspondant au nom du schéma à convertir. Choisissez ensuite ce schéma dans le panneau de gauche de votre projet. AWS SCT met en évidence le nom du schéma en bleu. Ouvrez le menu contextuel (clic droit) du schéma, puis choisissez Convertir le schéma, comme illustré ci-dessous.



Une fois que vous avez converti le schéma de votre base de données source, vous pouvez choisir des éléments de schéma dans le volet gauche de votre projet et afficher le schéma converti dans les panneaux centraux de votre projet. Le panneau situé dans la partie centrale inférieure affiche les propriétés et la commande SQL pour créer le schéma converti, comme illustré ci-après.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the project structure: Servers > SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > LARGE\_DB\_SS > Schemas [2] > dbo > Tables [4] > Account. The 'Account' table is selected. The main pane shows the SQL command for creating the table in the target Amazon RDS instance:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 )
14 )
15 ON [PRIMARY];

```

The interface also shows the 'Properties' tab for the table, with the 'SQL' tab selected. The cursor position is 0 and the table is modified. The target Amazon RDS instance is identified as 'Target Amazon RDS for MySQL table: Account'.

Une fois que vous avez converti votre schéma, vous pouvez enregistrer votre projet. Les informations de schéma de votre base de données source sont enregistrées avec votre projet. Cette fonctionnalité signifie que vous pouvez travailler hors ligne sans être connecté à votre base de données source. AWS SCT se connecte à votre base de données source pour mettre à jour le schéma dans votre projet si vous choisissez Refresh from Database (Actualiser depuis la base de données) pour votre base de données source. Pour de plus amples informations, veuillez consulter [Mise à jour et actualisation de votre schéma converti dans AWS SCT](#).

Vous pouvez créer un rapport d'évaluation de la migration de base de données des éléments qui ne peuvent pas être convertis automatiquement. Le rapport d'évaluation est utile pour identifier et résoudre les éléments de schéma qui ne peut pas être convertis automatiquement. Pour de plus amples informations, veuillez consulter [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#).

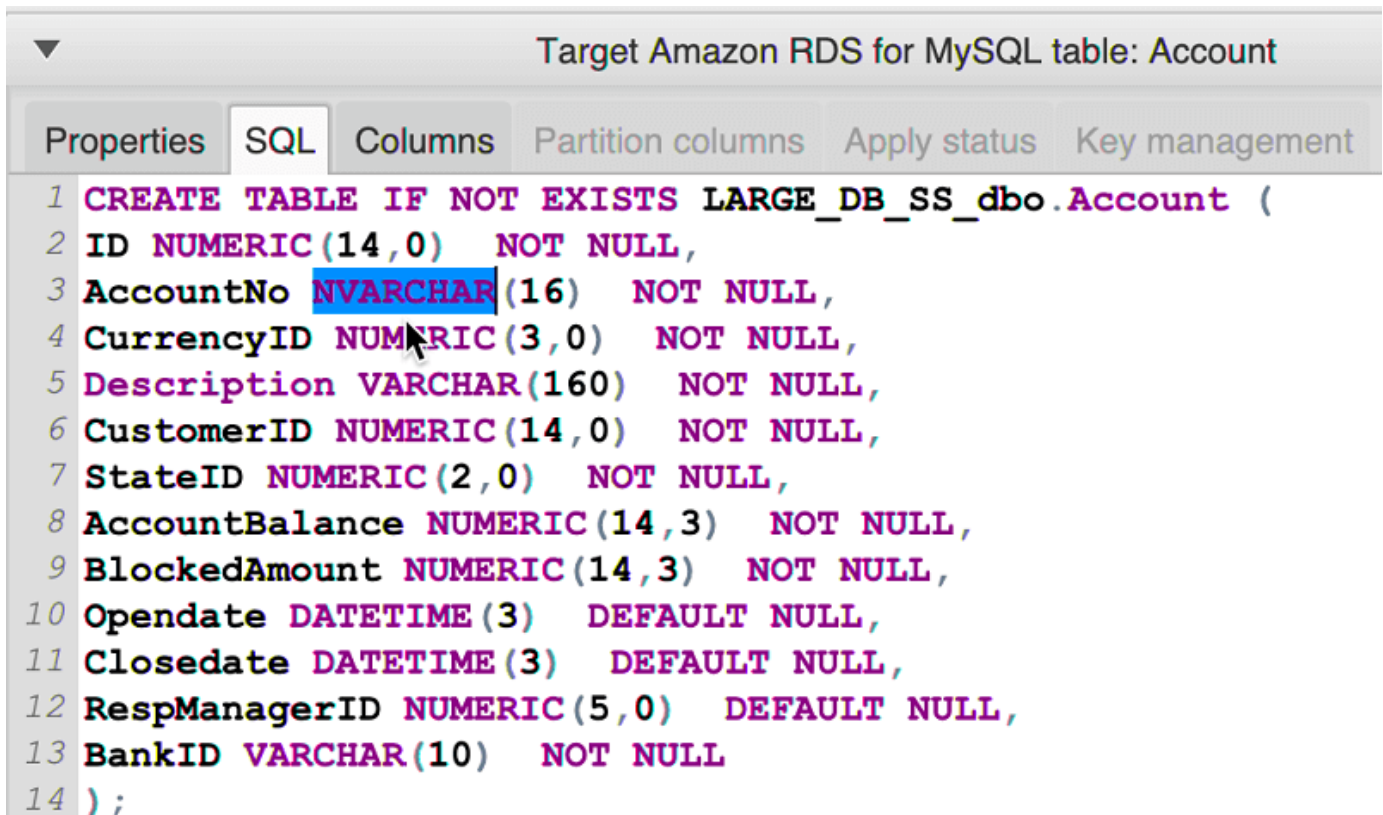
Lorsqu'il AWS SCT génère un schéma converti, il ne l'applique pas immédiatement à la base de données cible. En revanche, le schéma converti est stocké localement jusqu'à ce que vous soyez prêt à l'appliquer à la base de données cible. Pour de plus amples informations, veuillez consulter [Appliquer votre schéma converti](#).

## Modification du schéma converti

Vous pouvez modifier le schéma converti et enregistrer les modifications dans le cadre de votre projet.

Pour modifier le schéma converti

1. Dans le volet gauche qui affiche le schéma de votre base de données source, sélectionnez l'élément pour lequel vous voulez modifier le schéma converti.
2. Dans le volet situé dans la partie centrale inférieure qui affiche le schéma converti pour l'élément sélectionné, choisissez l'onglet SQL.
3. Dans le texte affiché pour l'onglet SQL, modifiez le schéma comme requis. Le schéma est automatiquement enregistré avec votre projet comme vous le mettez à jour.



```
▼ Target Amazon RDS for MySQL table: Account
Properties SQL Columns Partition columns Apply status Key management
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );
```

Les modifications que vous apportez au schéma converti sont stockées avec votre projet lorsque vous effectuez des mises à jour. Si vous convertissez nouvellement un élément de schéma de votre base de données source et que vous avez fait des mises à jour de schéma précédemment converti pour cet élément, ces mises à jour existantes sont remplacées par l'élément de schéma nouvellement converti basé sur votre base de données source.

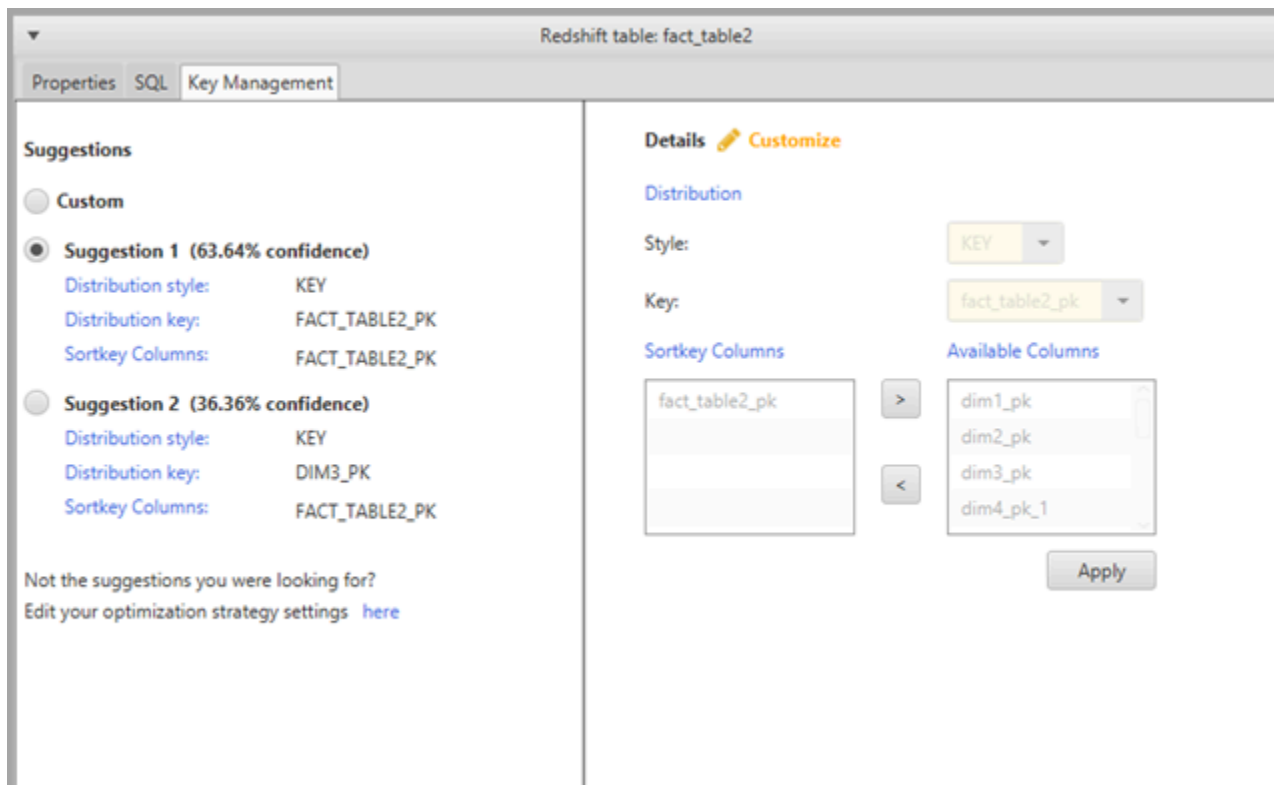
## Effacement d'un schéma converti

Tant que vous n'appliquez pas le schéma à votre base de données cible, le schéma converti AWS SCT n'est stocké que localement dans votre projet. Vous pouvez supprimer le schéma planifié de votre projet en choisissant le nœud de l'arborescence pour votre base de données cible, puis en choisissant Refresh from Database. Aucun schéma n'ayant été écrit dans votre base de données cible, l'actualisation à partir de la base de données supprime les éléments de schéma planifiés de votre AWS SCT projet pour qu'ils correspondent à ceux qui existent dans votre base de données cible.

## Gestion et personnalisation des clés dans AWS SCT

Après avoir converti votre schéma avec le AWS Schema Conversion Tool, vous pouvez gérer et modifier vos clés. La gestion des clés est le cœur d'une conversion d'entrepôt de données.

Pour gérer les clés, sélectionnez une table de votre base de données cible, puis choisissez l'onglet Key Management illustré ci-dessous.



Le volet de gauche contient des suggestions de clés et comprend l'évaluation de confiance pour chaque suggestion. Vous pouvez choisir une des suggestions, ou vous pouvez personnaliser la clé en l'éditant dans le volet droit.

Si les choix de la clé ne ressemblent pas à vos attentes, vous pouvez modifier vos stratégies d'optimisation et retenter la conversion. Pour de plus amples informations, veuillez consulter [Choix des stratégies et des règles d'optimisation à utiliser avec AWS SCT](#).

## Rubriques en relation

- [Choisissez la meilleure clé de tri](#)
- [Choisissez le meilleur style de distribution](#)

## Création et utilisation du rapport d'évaluation dans AWS SCT

AWS Schema Conversion Tool crée un rapport d'évaluation de la migration de base de données pour vous aider à convertir votre schéma. Le rapport d'évaluation de la migration de la base de données fournit des informations importantes à propos de la conversion du schéma de votre base de données source à votre base de données cible. Le rapport résume toutes les tâches de conversion de schéma

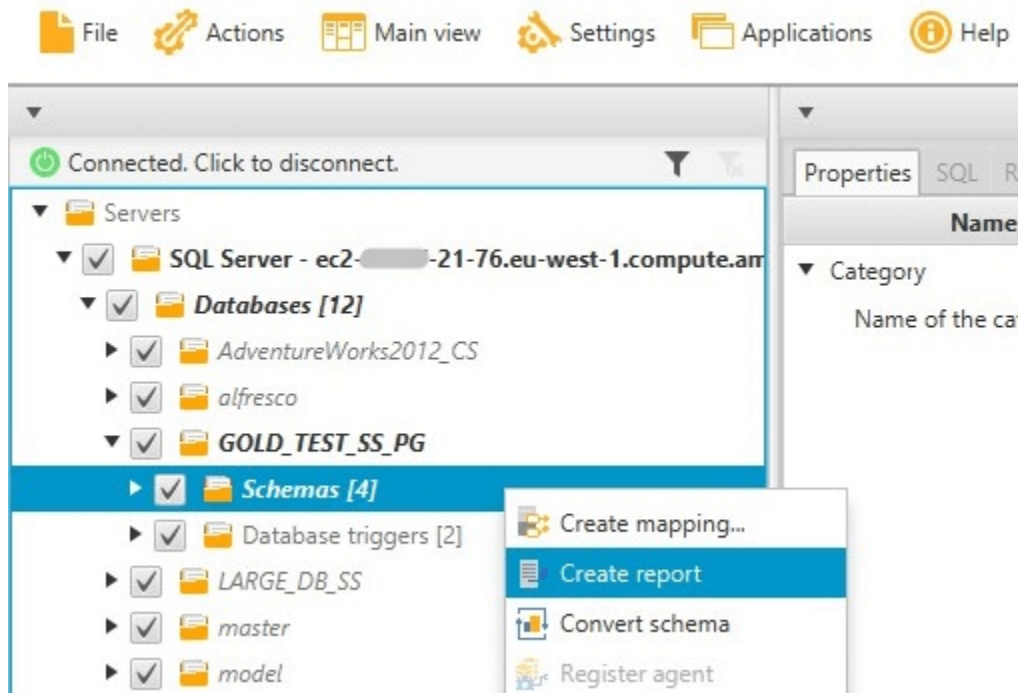
et détaille les éléments d'action de schéma qui ne peuvent pas être convertis vers le moteur de base de données de votre base de données cible. Ce rapport comprend également des estimations des efforts qui devront être fournis pour écrire le code équivalent dans votre base de données cible qui ne peuvent pas être converties automatiquement.

## Création d'un rapport d'évaluation de migration de base de données

Utilisez la procédure suivante pour créer un rapport d'évaluation de la migration de base de données.

Pour créer un rapport d'évaluation de migration de base de données

1. Dans le volet gauche qui affiche le schéma de votre base de données source, choisissez l'objet de schéma pour lequel vous souhaitez créer un rapport d'évaluation.
2. Ouvrez le menu contextuel (clic droit) pour l'objet, puis choisissez Create Report.



## Récapitulatif du rapport d'évaluation

Après avoir créé un rapport d'évaluation, l'affichage du rapport d'évaluation s'ouvre et affiche l'onglet Summary. L'onglet Summary affiche un récapitulatif du rapport d'évaluation de migration de base de données. Il indique les éléments qui ont été convertis automatiquement et ceux qui ne l'ont pas été.



Summary Action items

Save to CSV Save to PDF

## Database migration assessment report

Source database: GOLD\_TEST\_SS\_PG:21-76.eu-west-1.compute.amazonaws.com/GOLD\_TEST\_SS\_PG:1433  
 Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (X64) Mar 22 2021 18:10:24  
 Copyright (C) 2019 Microsoft Corporation  
 Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763:) (Hypervisor)  
 Case sensitivity: OFF

### Executive summary

We completed the analysis of your Microsoft SQL Server source database and estimate that 90% of the database storage objects and 77% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for PostgreSQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, table types, sequences, synonyms and xml schema collections. Database code objects include triggers, views, procedures, scalar functions, inline functions, table-valued functions and database triggers. Based on the source code syntax analysis, we estimate 94% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically. To complete the migration, we recommend 3,300 conversion action(s) ranging from simple tasks to medium-complexity actions to complex conversion actions.

Migration guidance for database objects that could not be converted automatically can be found [here](#).

### Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

**Figure: Conversion statistics for database storage objects**

Object Type	Count	Automatically converted	Simple actions	Medium-complexity actions	Complex actions
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	0%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	35%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	0%	80%	0%

En ce qui concerne les éléments du schéma qui ne peuvent pas être convertis automatiquement pour le moteur de base de données cible, ce récapitulatif comprend une estimation de l'effort nécessaire pour créer des éléments de schéma dans votre instance DB cible équivalents à ceux présents dans votre source.

Ce rapport classe le temps estimé pour convertir ces éléments de schéma de la façon suivante :

- Simple : actions qui peuvent être effectuées en moins d'une heure.
- Moyen — Actions plus complexes pouvant être réalisées en une à quatre heures.
- Important — Actions très complexes qui prennent plus de quatre heures.

## Éléments d'action du rapport d'évaluation

L'affichage de rapport d'évaluation comprend également un onglet Action Items. Cet onglet contient une liste des éléments qui ne peuvent pas être convertis automatiquement pour le moteur de base de données de votre base de données cible. Si vous sélectionnez un élément d'action dans la liste, AWS SCT met en évidence l'élément de votre schéma auquel s'applique l'élément d'action.

Le rapport contient également des recommandations sur la manière de convertir manuellement l'élément de schéma. Pour plus d'informations sur la façon de gérer les conversions manuelles, consultez [Gestion des conversions manuelles dans AWS SCT](#).

The screenshot shows the AWS Schema Conversion Tool interface. The top menu includes File, Actions, Assessment Report view, Settings, Applications, Help, Add source, and Add target. The main window is divided into several sections:

- Summary / Action items:** A tabbed interface with 'Action items' selected.
- Source:** A tree view showing the source database structure, including Servers, Databases [12], and Schemas [1]. The selected schema is 'dbo'.
- Issues List:** A list of issues with details such as the issue number, description, recommended action, and number of occurrences. Issues include:
  - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required.
  - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option.
  - Issue 794:** MySQL doesn't support user-defined data types.
  - Issue 826:** Check the default value for a DateTime variable.
  - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values.
  - Issue 9997:** Unable to resolve objects.
  - Issue 690:** MySQL doesn't support table types.
  - Issue 811:** Unable to convert functions.
- Issue Details (Issue 609):** A detailed view of the selected issue, showing the recommended action: 'Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table.' It also shows the source SQL code for the procedure 'POSITION\_UPDATE\_CASH\_CGT\_BULK'.
- Target:** A table showing the target Amazon RDS schema details, including the category 'Schemas'.

## Sauvegarde du rapport d'évaluation

Vous pouvez enregistrer une copie locale du rapport d'évaluation de la migration de la base de données dans un fichier PDF ou un fichier de valeurs séparées par des virgules (CSV). Le fichier CSV contient uniquement des informations sur l'élément d'action. Le fichier PDF contient les informations sur le récapitulatif et les éléments d'action, comme illustré dans l'exemple suivant.

## Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

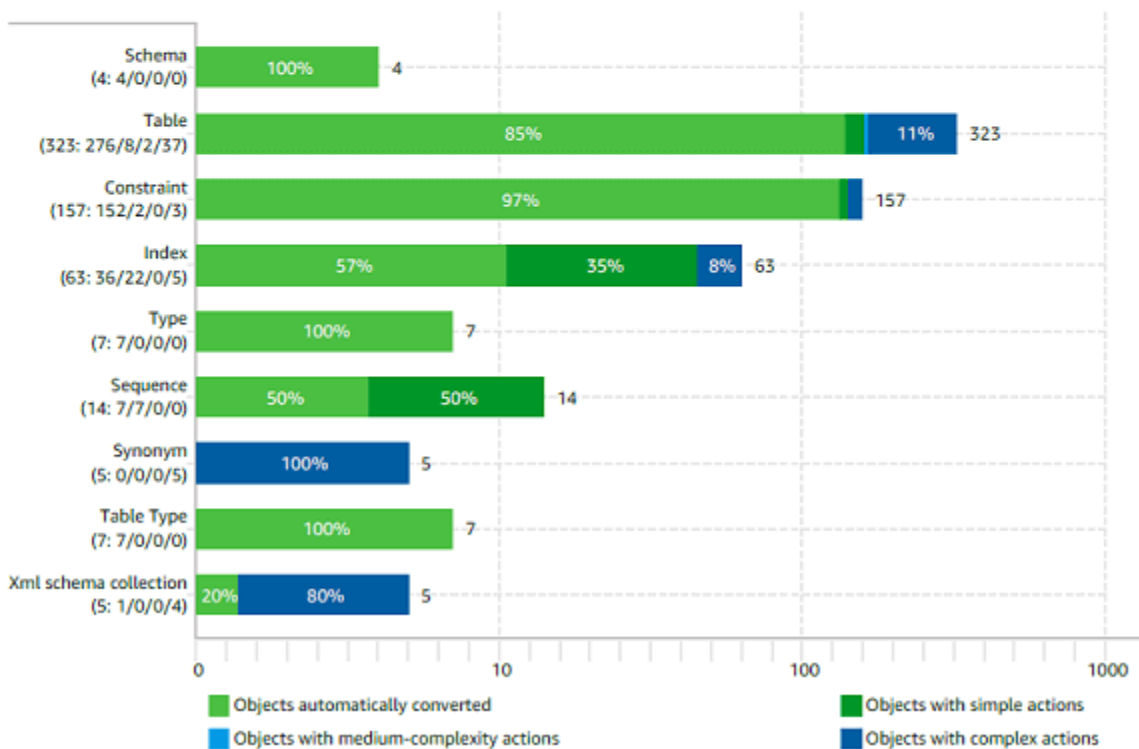
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects



## Gestion des conversions manuelles dans AWS SCT

Le rapport d'évaluation inclut une liste des éléments qui ne peuvent pas être convertis automatiquement pour le moteur de base de données de votre base de données cible. Pour chaque élément qui ne peut pas être converti, il y a un élément de l'action sur l'onglet Action Items.

Vous pouvez réagir aux éléments d'action du rapport d'évaluation des manières suivantes :

- Modifiez votre schéma de base de données source.

- Modifiez votre schéma de base de données cible.

## Modification de votre schéma source

Pour certains éléments, il peut être plus facile de modifier le schéma de base de données dans votre base de données source en un schéma qui peut être converti automatiquement. Tout d'abord, vérifiez que les nouvelles modifications sont compatibles avec l'architecture de votre application, puis mettez à jour le schéma dans votre base de données source. Enfin, actualisez votre projet avec les informations de schéma mises à jour. Vous pouvez ensuite convertir le schéma mis à jour et générer un nouveau rapport d'évaluation de la migration de la base de données. Les éléments d'action n'apparaissent plus pour les éléments qui ont été modifiés dans le schéma source.

L'avantage de ce processus est que le schéma mis à jour est toujours disponible lors de l'actualisation à parti de votre base de données source.

## Modification de votre schéma cible

Pour certains éléments, il peut être plus facile d'appliquer le schéma converti à votre base de données cible, et d'ajouter ensuite les éléments de schéma équivalent manuellement à votre base de données cible pour les éléments qui n'ont pas pu être convertis automatiquement. Vous pouvez écrire tout le schéma qui peut être converti automatiquement pour votre base de données cible en appliquant le schéma. Pour de plus amples informations, veuillez consulter [Enregistrer et appliquer votre schéma converti dans AWS SCT](#).

Le schéma qui est écrit dans votre base de données cible ne contient pas les éléments qui ne peuvent pas être convertis automatiquement. Après avoir appliqué le schéma à votre base de données cible, vous pouvez alors créer manuellement un schéma dans votre base de données cible qui est équivalent à celui de la base de données source. Les éléments d'action dans le rapport d'évaluation de la migration de base de données contiennent des suggestions sur la façon de créer le schéma équivalent.

### Warning

Si vous créez manuellement le schéma de votre base de données cible, enregistrez une copie de n'importe quel travail manuel que vous faites. Si vous appliquez de nouveau le schéma converti à partir de votre projet à votre base de données cible, il remplace le travail manuel que vous avez terminé.

Dans certains cas, vous ne pouvez pas créer de schéma équivalent dans votre base de données cible. Vous devrez reconcevoir une partie de votre application et de votre base de données afin d'utiliser les fonctionnalités disponibles dans le moteur de base de données pour votre base de données cible. Dans d'autres cas, vous pourrez simplement ignorer le schéma qui ne peut pas être converti automatiquement.

## Mise à jour et actualisation de votre schéma converti dans AWS SCT

Vous pouvez mettre à jour le schéma source et le schéma cible de votre AWS Schema Conversion Tool projet.

- **Source** : si vous mettez à jour le schéma de votre base de données source, AWS SCT remplace le schéma de votre projet par le schéma le plus récent de votre base de données source. A l'aide de cette fonctionnalité, vous pouvez mettre à jour votre projet si les modifications ont été apportées au schéma de votre base de données source.
- **Cible** : si vous mettez à jour le schéma de votre base de données cible AWS SCT, remplacez le schéma de votre projet par le schéma le plus récent de votre base de données cible. Si vous n'avez appliqué aucun schéma à votre base de données cible, AWS SCT efface le schéma converti de votre projet. Vous pouvez ensuite convertir le schéma de votre base de données source en une base de données cible propre.

Pour mettre à jour le schéma de votre AWS SCT projet, sélectionnez Actualiser depuis la base de données.

## Enregistrer et appliquer votre schéma converti dans AWS SCT

Lorsque le schéma converti est AWS Schema Conversion Tool généré (comme indiqué dans [Conversion de votre schéma à l'aide de AWS SCT](#)), il n'applique pas immédiatement le schéma converti à la base de données cible. Au lieu de cela, le schéma converti est stocké localement dans votre projet jusqu'à ce que vous soyez prêt à l'appliquer à la base de données cible. A l'aide de cette fonctionnalité, vous pouvez travailler avec des éléments de schéma qui ne peuvent pas être convertis automatiquement pour votre moteur de base de données cible. Pour plus d'informations sur les éléments qui ne peuvent pas être convertis automatiquement, consultez [À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool](#).

De manière facultative, l'outil peut enregistrer votre schéma converti en un fichier en tant que script SQL avant d'appliquer le schéma à votre base de données cible. Vous pouvez également faire que l'outil applique le schéma converti directement à votre base de données cible.

## Enregistrement de votre schéma converti dans un fichier

Vous pouvez enregistrer votre schéma converti sous forme de scripts SQL dans un fichier texte. En utilisant cette approche, vous pouvez modifier les scripts SQL générés AWS SCT pour traiter les éléments que l'outil ne peut pas convertir automatiquement. Vous pouvez ensuite exécuter vos scripts mis à jour sur votre instance DB cible pour appliquer votre schéma converti à votre base de données cible.

Pour enregistrer votre schéma converti sous forme de scripts SQL

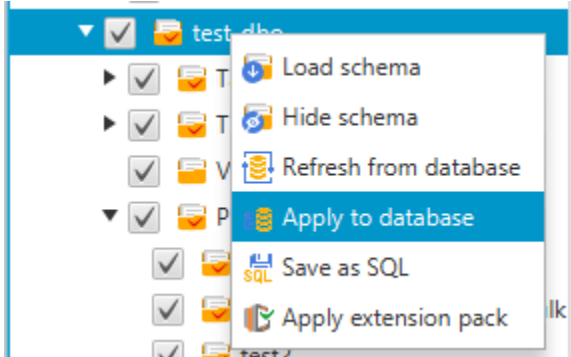
1. Choisissez votre schéma et ouvrez le menu contextuel (clic droit).
2. Choisissez Enregistrer en tant que SQL.
3. Entrez le nom du fichier et choisissez Enregistrer.
4. Enregistrez votre schéma converti à l'aide de l'une des options suivantes :
  - Fichier unique
  - Un seul fichier par étape
  - Un seul fichier par relevé

Pour choisir le format du script SQL

1. Dans le menu Paramètres, choisissez Paramètres du projet.
2. Choisissez Enregistrer les scripts.
3. Pour Vendor, choisissez la plate-forme de base de données.
4. Pour Enregistrer les scripts SQL sur, choisissez la manière dont vous souhaitez enregistrer le script de schéma de base de données.
5. Cliquez sur OK pour enregistrer les paramètres.

## Appliquer votre schéma converti

Une fois que vous êtes prêt à appliquer votre schéma converti à votre base de données cible, choisissez l'élément de schéma dans le volet droit du projet. Ouvrez le menu contextuel (clic droit) pour l'élément de schéma, puis choisissez **Apply to database**, comme indiqué ci-dessous.



## Le schéma du pack d'extension

La première fois que vous appliquez votre schéma converti à votre instance de base de données cible, AWS SCT ajoute un schéma supplémentaire à cette dernière. Ce schéma met en œuvre les fonctions système de la base de données source nécessaires lors de l'écriture du schéma converti dans l'instance DB cible. Le schéma est appelé schéma de kit d'extension.

Ne modifiez pas ce schéma ou vous risquez d'obtenir des résultats inattendus dans le schéma converti écrit dans l'instance de base de données cible. Lorsque votre schéma est entièrement migré vers votre instance de base de données cible et que vous n'en avez plus besoin AWS SCT, vous pouvez supprimer le schéma du pack d'extension.

Le schéma de kit d'extension est nommé selon votre base de données source, comme suit :

- Greenplum : `aws_greenplum_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- Netezza : `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Flocon de neige : `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertica : `aws_vertica_ext`

Pour de plus amples informations, veuillez consulter [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

## Bibliothèques Python

Pour créer des fonctions personnalisées dans Amazon Redshift, vous devez utiliser le langage Python. Utilisez le pack d' AWS SCT extension pour installer des bibliothèques Python pour votre base de données Amazon Redshift. Pour de plus amples informations, veuillez consulter [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

## Conversion de données depuis Amazon Redshift à l'aide de AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool pour optimiser votre base de données Amazon Redshift. En utilisant votre base de données Amazon Redshift comme source et une base de données Amazon Redshift de test comme cible AWS SCT , vous pouvez recommander des clés de tri et des clés de distribution pour optimiser votre base de données.

## Optimisation de votre base de données Amazon Redshift

Utilisez la procédure suivante pour optimiser votre base de données Amazon Redshift.

Pour optimiser votre base de données Amazon Redshift

1. Prenez un instantané manuel de votre cluster Amazon Redshift à titre de sauvegarde. Vous pouvez supprimer l'instantané une fois que vous avez terminé d'optimiser votre cluster Amazon Redshift et de tester les modifications que vous apportez. Pour plus d'informations, consultez la section [Instantanés Amazon Redshift](#).
2. Choisissez l'objet de schéma à convertir dans le volet gauche de votre projet. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Collect Statistics.

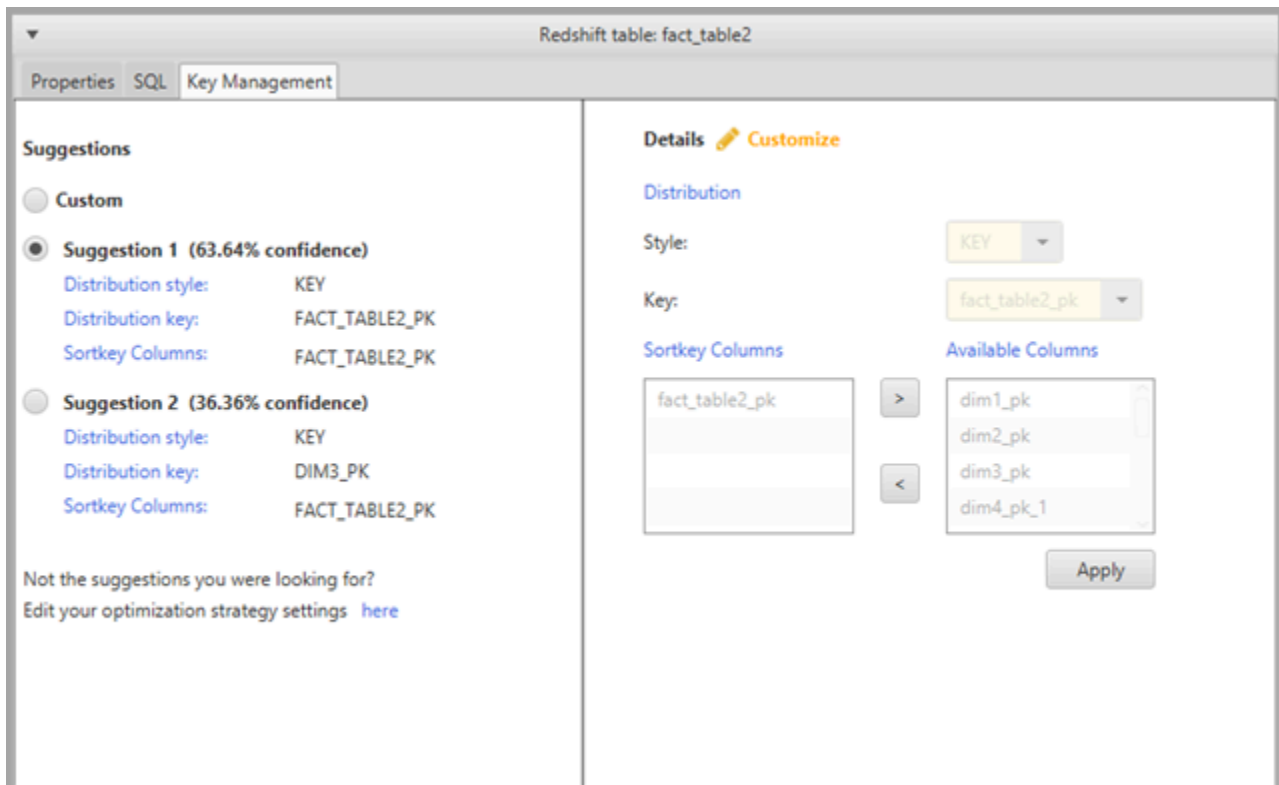
AWS SCT utilise les statistiques pour suggérer des clés de tri et des clés de distribution.

3. Choisissez l'objet de schéma à optimiser dans le volet gauche de votre projet. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Run Optimization.

AWS SCT fait des suggestions pour les clés de tri et les clés de distribution.



4. Pour passer en revue les suggestions, développez le nœud de tables sous votre schéma dans le volet gauche de votre projet, puis choisissez une table. Choisissez l'onglet Key Management comme illustré ci-après.



Le volet de gauche contient des suggestions de clés et comprend l'évaluation de confiance pour chaque suggestion. Vous pouvez choisir une des suggestions, ou vous pouvez personnaliser la clé en l'éditant dans le volet droit.

5. Vous pouvez créer un rapport contenant les suggestions d'optimisation. Pour créer le rapport, procédez comme suit :
  - a. Choisissez l'objet de schéma que vous avez optimisé dans le volet gauche de votre projet. Ouvrez le menu contextuel (clic droit) pour l'objet, puis choisissez Create Report.

Le rapport s'ouvre dans la fenêtre principale et l'onglet Summary s'affiche. Le nombre d'objets comportant des suggestions d'optimisation s'affiche dans le rapport.
  - b. Choisissez l'onglet Action Items pour afficher les suggestions de clés sous forme de rapport.
  - c. Vous pouvez enregistrer une copie locale du rapport d'optimisation dans un fichier PDF ou un fichier de valeurs séparées par des virgules (CSV). Le fichier CSV contient uniquement des informations sur l'élément d'action. Le fichier PDF contient à la fois le récapitulatif et les informations sur les éléments d'action.

6. Pour appliquer les optimisations proposées pour votre base de données, choisissez un objet dans le volet droit de votre projet. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez **Apply to database**.

# Conversion de données à l'aide de processus ETL dans AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour migrer les processus d'extraction, de transformation et de chargement (ETL). Ce type de migration inclut la conversion de la logique métier liée à l'ETL. Cette logique peut résider dans vos entrepôts de données sources ou dans des scripts externes que vous exécutez séparément.

Actuellement, AWS SCT prend en charge la conversion de scripts ETL en AWS Glue objets vers Amazon Redshift RSQL, comme indiqué dans le tableau suivant.

Source	Cible
Scripts ETL Informatica	Informatica
Packages ETL Microsoft SQL Server Integration Services (SSIS)	AWS Glue ou AWS Glue Studio
Scripts Shell avec commandes intégrées à partir de Teradata Basic Teradata Query (BTEQ)	Amazon Redshift RSQL
Scripts ETL Teradata BTEQ	AWS Glue ou Amazon Redshift RSQL
Scripts de FastExport tâches Teradata	Amazon Redshift RSQL
Scripts de FastLoad tâches Teradata	Amazon Redshift RSQL
Scripts de MultiLoad tâches Teradata	Amazon Redshift RSQL

## Rubriques

- [Conversion des processus ETL en AWS Glue in AWS Schema Conversion Tool](#)
- [Conversion des processus ETL en AWS Glue in AWS Schema Conversion Tool](#)
- [Conversion de scripts ETL Informatica avec AWS Schema Conversion Tool](#)
- [Conversion de SSIS en AWS Glue with AWS SCT](#)

- [Conversion de packages SSIS en with AWS Glue StudioAWS Schema Conversion Tool](#)
- [Conversion de scripts Teradata BTEQ en Amazon Redshift RSQL avec AWS SCT](#)
- [Conversion de scripts shell contenant des commandes BTEQ intégrées en Amazon Redshift RSQL avec AWS Schema Conversion Tool](#)
- [Conversion de FastExport scripts vers Amazon Redshift RSQL avec AWS Schema Conversion Tool](#)
- [Conversion de scripts de FastLoad travail en Amazon Redshift RSQL avec AWS Schema Conversion Tool](#)
- [Conversion de MultiLoad scripts vers Amazon Redshift RSQL avec AWS Schema Conversion Tool](#)

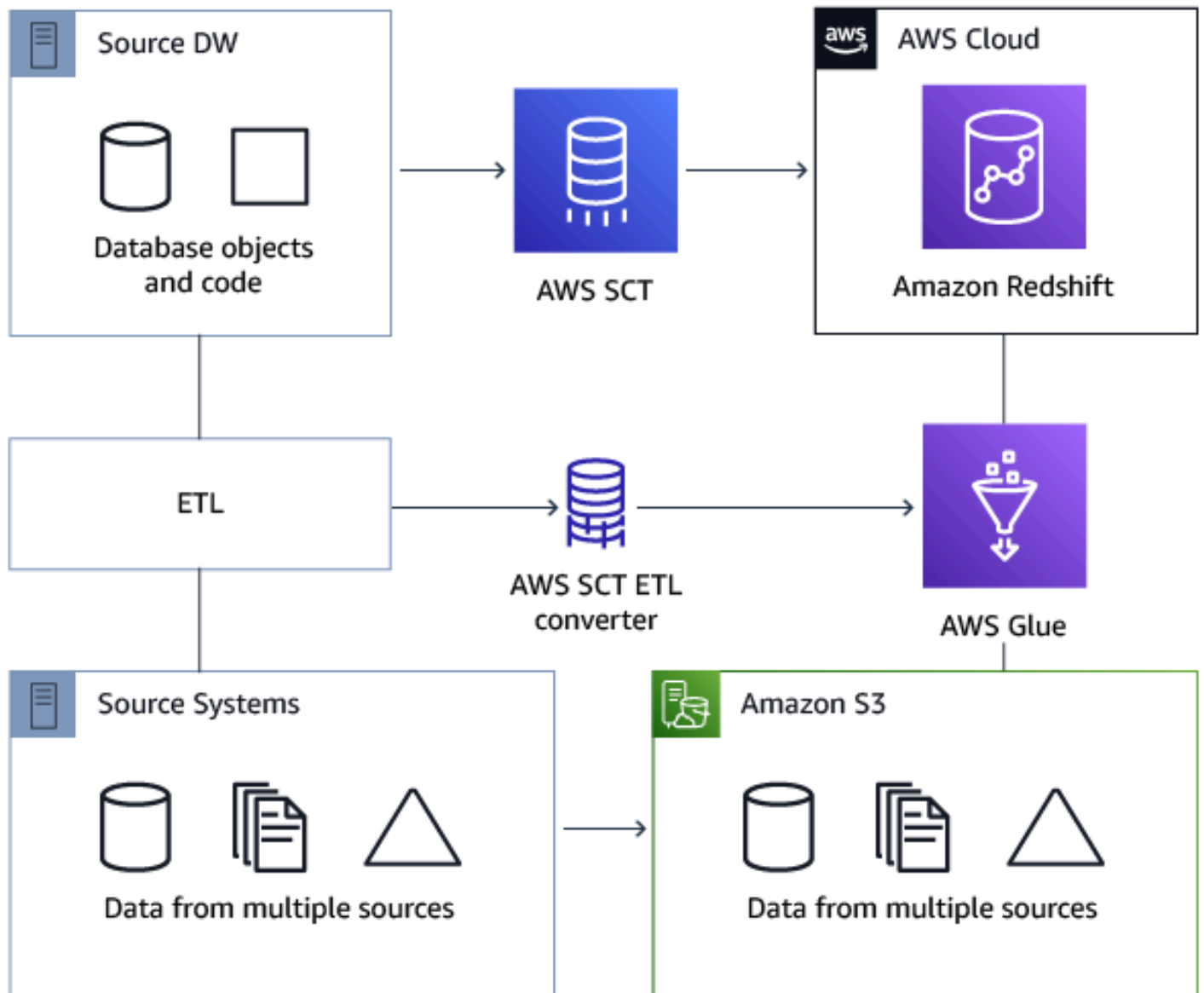
## Conversion des processus ETL en AWS Glue in AWS Schema Conversion Tool

Vous trouverez ci-dessous un aperçu du processus de conversion des scripts ETL en AWS Glue with AWS SCT. Dans cet exemple, nous convertissons une base de données Oracle en Amazon Redshift, ainsi que les processus ETL utilisés avec les bases de données sources et les entrepôts de données.

### Rubriques

- [Prérequis](#)
- [Comprendre le catalogue AWS Glue de données](#)
- [Limitations relatives à la conversion à l' AWS SCT aide de AWS Glue](#)
- [Étape 1 : Créer un projet](#)
- [Étape 2 : Création d'un AWS Glue emploi](#)

Le schéma d'architecture suivant montre un exemple de projet de migration de base de données qui inclut la conversion de scripts ETL en AWS Glue.



## Prérequis

Avant de commencer, vous devez exécuter les actions suivantes :

- Migrez toutes les bases de données source vers lesquelles vous souhaitez migrer AWS.
- Migrez les entrepôts de données cibles vers AWS.
- Établir une liste de tous les codes impliqués dans votre processus ETL.
- Établir une liste de toutes les informations de connexion nécessaires pour chaque base de données.

Il a également AWS Glue besoin d'autorisations pour accéder à d'autres AWS ressources en votre nom. Vous fournissez ces autorisations en utilisant AWS Identity and Access Management (IAM). Assurez-vous d'avoir créé une politique IAM pour AWS Glue. Pour plus d'informations, consultez la section [Création d'une politique IAM pour le AWS Glueservice](#) dans le guide du AWS Glue développeur.

## Comprendre le catalogue AWS Glue de données

Dans le cadre du processus de conversion, AWS Glue charge les informations concernant les bases de données source et cible. Il organise ces informations en catégories, dans une structure appelée arbre. Cette structure comprend les éléments suivants :

- Connexions — Paramètres de connexion
- Crawlers : liste des robots d'exploration, un robot pour chaque schéma
- Bases de données — Conteneurs contenant des tables
- Tables — Définitions de métadonnées qui représentent les données contenues dans les tables
- Tâches ETL — Logique métier qui exécute le travail ETL
- Déclencheurs : logique qui contrôle le moment où une tâche ETL s'exécute AWS Glue (que ce soit à la demande, selon un calendrier ou déclenchée par des événements de tâche)

Le AWS Glue catalogue de données est un index de l'emplacement, du schéma et des métriques d'exécution de vos données. Lorsque vous travaillez avec AWS Glue et AWS SCT, le catalogue de AWS Glue données contient des références à des données utilisées comme sources et cibles de vos tâches ETL dans AWS Glue. Pour créer votre entrepôt de données, cataloguez ces données.

Les informations du catalogue de données vous permettent de créer et de surveiller vos tâches ETL. En règle générale, vous devez exécuter un crawler pour effectuer l'inventaire des données de vos magasins de données, mais il y a d'autres manières d'ajouter des tables de métadonnées à votre catalogue de données.

Lorsque vous définissez une table dans votre catalogue de données, vous l'ajoutez à une base de données. Une base de données est utilisée pour organiser les tables dans AWS Glue.

## Limitations relatives à la conversion à l' AWS SCT aide de AWS Glue

Les limitations suivantes s'appliquent lors de la conversion à l'aide AWS SCT de with AWS Glue.

Ressource	Limite par défaut
Nombre de bases de données pour chaque compte	10 000
Nombre de tables pour chaque base de données	100 000
Nombre de partitions pour chaque table	1 000 000
Nombre de versions de table pour chaque table	100 000
Nombre de tables pour chaque compte	1 000 000
Nombre de partitions pour chaque compte	10 000 000
Nombre de versions de table pour chaque compte	1 000 000
Nombre de connexions pour chaque compte	1 000
Nombre d'analyseurs par compte	25
Nombre de tâches pour chaque compte	25
Nombre de déclencheurs pour chaque compte	25
Nombre d'exécutions de tâches simultanées par compte	30
Nombre d'exécutions de tâches simultanées pour chaque tâche	3
Nombre de tâches pour chaque déclencheur	10
Nombre de points de terminaison de développement par compte	5

Nombre maximal d'unités de traitement de données (DPUs) utilisées par un terminal de développement à la fois	5
Maximum DPUs utilisé par un rôle à la fois	100
Longueur du nom de la base de données	Illimité  Pour la compatibilité avec d'autres magasins de métadonnées, tels qu'Apache Hive, le nom est modifié de façon à utiliser des lettres minuscules.  Si vous prévoyez d'accéder à la base de données depuis Amazon Athena, indiquez un nom composé uniquement de caractères alphanumériques et de soulignement.
Longueur du nom de la connexion	Illimité
Longueur du nom de l'analyseur	Illimité

## Étape 1 : Créer un projet

Pour créer un nouveau projet, suivez ces étapes de haut niveau :

1. Créez un nouveau projet dans AWS SCT. Pour de plus amples informations, veuillez consulter [Démarrage et gestion de projets dans AWS SCT](#).
2. Ajoutez vos bases de données source et cible au projet. Pour de plus amples informations, veuillez consulter [Ajouter des serveurs au projet dans AWS SCT](#).

Assurez-vous d'avoir sélectionné Utiliser AWS Glue dans les paramètres de connexion à la base de données cible. Pour ce faire, choisissez l'AWS Glueonglet. Pour Copier depuis le AWS profil, choisissez le profil que vous souhaitez utiliser. Le profil doit automatiquement renseigner la clé AWS d'accès, la clé secrète et le dossier du compartiment Amazon S3. Si ce n'est pas le cas, saisissez ces informations vous-même. Après avoir sélectionné OK, AWS Glue analyse les objets et charge les métadonnées dans le catalogue de AWS Glue données.



En fonction de vos paramètres de sécurité, il est possible qu'un message d'avertissement s'affiche pour indiquer que votre compte ne dispose pas de privilèges suffisants pour certains des schémas sur le serveur. Si vous avez accès aux schémas que vous utilisez, vous pouvez ignorer sans risque ce message.

3. Pour terminer la préparation de l'importation de votre ETL, connectez-vous à vos bases de données source et cible. Pour ce faire, choisissez votre base de données dans l'arborescence des métadonnées source ou cible, puis sélectionnez **Connect to the server**.

AWS Glue crée une base de données sur le serveur de base de données source et une autre sur le serveur de base de données cible pour faciliter la conversion ETL. La base de données du serveur cible contient le catalogue de AWS Glue données. Pour rechercher des objets spécifiques, utilisez la fonction de recherche dans les panneaux source ou cible.

Pour voir comment un objet spécifique est converti, recherchez un élément à convertir et choisissez **Convertir le schéma** dans son menu contextuel (clic droit). AWS SCT transforme cet objet sélectionné en script.

Vous pouvez consulter le script converti dans le dossier **Scripts** du panneau de droite. Actuellement, le script est un objet virtuel, disponible uniquement dans le cadre de votre AWS SCT projet.

Pour créer une AWS Glue tâche avec votre script converti, téléchargez-le sur Amazon S3. Pour télécharger le script sur Amazon S3, choisissez-le, puis sélectionnez **Enregistrer dans S3** dans son menu contextuel (clic droit).

## Étape 2 : Création d'un AWS Glue emploi

Après avoir enregistré le script sur Amazon S3, vous pouvez le choisir, puis choisir **Configure AWS Glue Job** pour ouvrir l'assistant de configuration de la AWS Glue tâche. L'assistant facilite cette configuration :

1. Dans le premier onglet de l'assistant, **Design Data Flow**, vous pouvez choisir une stratégie d'exécution et la liste des scripts que vous souhaitez inclure dans cette tâche. Vous pouvez choisir des paramètres pour chaque script. Vous pouvez également réorganiser les scripts afin qu'ils s'exécutent dans l'ordre correct.
2. Dans le deuxième onglet, vous pouvez nommer votre tâche et configurer directement les paramètres pour AWS Glue. Vous pouvez configurer les paramètres suivants sur cet écran :
  - **AWS Identity and Access Management Rôle (IAM)**

- Noms et chemins des fichiers de script
  - Chiffrez le script à l'aide du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
  - Répertoire temporaire
  - Chemin de la bibliothèque Python générée
  - Chemin de la bibliothèque Python de l'utilisateur
  - Chemin pour les fichiers .jar dépendants
  - Chemin de fichiers référencés
  - Simultané DPU pour chaque tâche exécutée
  - Simultanéité maximum
  - Délai d'expiration de la tâche en minutes.
  - Seuil de notification de délai (minutes)
  - Nombre de nouvelles tentatives
  - Configuration de la sécurité
  - Chiffrement côté serveur
3. À la troisième étape, ou dans le troisième onglet, vous choisissez la connexion configurée vers le point de terminaison cible.

Une fois que vous avez terminé de configurer la tâche, elle s'affiche sous les tâches ETL dans le catalogue de AWS Glue données. Si vous choisissez la tâche, les paramètres s'affichent afin que vous puissiez les examiner ou les modifier. Pour créer une nouvelle tâche dans AWS Glue, choisissez Create AWS Glue Job dans le menu contextuel (clic droit) de la tâche. Cette opération a pour effet d'appliquer la définition du schéma. Pour actualiser l'affichage, choisissez Refresh from database (Actualiser à partir de la base de données) dans le menu contextuel (clic droit).

À ce stade, vous pouvez consulter votre tâche dans la AWS Glue console. Pour ce faire, connectez-vous à la AWS Glue console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/glue/>.

Vous pouvez tester la nouvelle tâche pour vous assurer qu'elle fonctionne correctement. Vérifiez les données de votre table source, puis assurez-vous que la table cible est vide. Exécutez la tâche et vérifiez à nouveau. Vous pouvez consulter les journaux d'erreurs depuis la AWS Glue console.

# Conversion des processus ETL en AWS Glue in AWS Schema Conversion Tool

Dans les sections suivantes, vous trouverez une description d'une conversion qui appelle des opérations AWS Glue d'API en Python. Pour plus d'informations, consultez [Program AWS Glue ETL scripts in Python](#) dans le manuel du AWS Glue développeur.

## Rubriques

- [Étape 1 : Créer une base de données](#)
- [Étape 2 : Création d'une connexion](#)
- [Étape 3 : créer un AWS Glue crawler](#)

## Étape 1 : Créer une base de données

La première étape consiste à créer une nouvelle base de données dans un catalogue de AWS Glue données à l'aide de l'[API du AWS SDK](#). Lorsque vous définissez une table dans le catalogue de données, vous l'ajoutez à une base de données. Une base de données est utilisée pour organiser les tables dans AWS Glue.

L'exemple suivant illustre la `create_database` méthode de l'API Python pour AWS Glue.

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Si vous utilisez Amazon Redshift, le nom de la base de données est formé comme suit.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Le nom complet du cluster Amazon Redshift pour cet exemple est le suivant.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Vous trouverez ci-dessous un exemple de nom de base de données bien formé. Dans ce cas, `rsdbb03` est le nom, qui est la première partie du nom complet du point de terminaison du cluster. La base de données est nommée `dev` et le schéma est `ora_glue`.

```
rsdbb03_dev_ora_glue
```

## Étape 2 : Création d'une connexion

Créez une nouvelle connexion dans un catalogue de données à l'aide de l'[API du AWS SDK](#).

L'exemple suivant illustre l'utilisation de la [create\\_connection](#) méthode de l'API Python pour AWS Glue.

```
response = client.create_connection(  
    ConnectionInput={  
        'Name': 'Redshift_abcde03.aabbcc112233.us-west-2.redshift.amazonaws.com_dev',  
        'Description': 'Created from SCT',  
        'ConnectionType': 'JDBC',  
        'ConnectionProperties': {  
            'JDBC_CONNECTION_URL': 'jdbc:redshift://aabbcc03.aabbcc112233.us-  
west-2.redshift.amazonaws.com:5439/dev',  
            'USERNAME': 'user_name',  
            'PASSWORD': 'password'  
        },  
        'PhysicalConnectionRequirements': {  
            'AvailabilityZone': 'us-west-2c',  
            'SubnetId': 'subnet-a1b23c45',  
            'SecurityGroupIdList': [  
                'sg-000a2b3c', 'sg-1a230b4c', 'sg-aba12c3d', 'sg-1abb2345'  
            ]  
        }  
    }  
)
```

Les paramètres utilisés dans `create_connection` sont les suivants :

- **Name**(chaîne UTF-8) — obligatoire. Pour Amazon Redshift, le nom de connexion est formé comme suit :`Redshift_<Endpoint-name>_<redshift-database-name>`, par exemple : `Redshift_abcde03_dev`
- **Description**(chaîne UTF-8) — Votre description de la connexion.
- **ConnectionType**(chaîne UTF-8) — Obligatoire. Type de connexion. À l'heure actuelle, seul JDBC est pris en charge ; SFTP n'est pas pris en charge.
- **ConnectionProperties**(dict) — Obligatoire. Liste des paires clé-valeur utilisées comme paramètres pour cette connexion, y compris l'URL de connexion JDBC, le nom d'utilisateur et le mot de passe.
- **PhysicalConnectionRequirements**(dict) — Exigences de connexion physique, notamment les suivantes :
  - **SubnetId**(chaîne UTF-8) — L'ID du sous-réseau utilisé par la connexion.
  - **SecurityGroupIdList**(liste) : liste d'identifiants de groupes de sécurité utilisée par la connexion.
  - **AvailabilityZone**(chaîne UTF-8) — Obligatoire. Zone de disponibilité qui contient le point de terminaison. Ce paramètre est obsolète.

## Étape 3 : créer un AWS Glue crawler

Ensuite, vous créez un AWS Glue robot d'exploration pour remplir le AWS Glue catalogue. Pour plus d'informations, consultez la section [Catalogage de tables à l'aide d'un robot d'exploration](#) dans le Guide du AWS Glue développeur.

La première étape de l'ajout d'un robot d'exploration consiste à créer une nouvelle base de données dans un catalogue de données à l'aide de l'API du [AWS SDK](#). Avant de commencer, assurez-vous de supprimer d'abord toute version précédente de celui-ci en utilisant l'`delete_crawler` opération.

Lorsque vous créez votre analyseur, quelques considérations s'appliquent :

- Pour le nom de l'analyseur, utilisez le format `<redshift_node_name>_<redshift_database_name>_<redshift_shema_name>`, par exemple : `abcde03_dev_ora_glue`
- Utilisez un rôle IAM qui existe déjà. Pour plus d'informations sur la création de rôles IAM, consultez la section [Création de rôles IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez le nom du rôle de métrique que vous avez créé à l'étape précédente.

- Utilisez le paramètre `ConnectionName`, qui est obligatoire.
- Pour le paramètre `path`, utilisez le chemin d'accès à la cible JDBC, par exemple : `dev/ora_glue/%`

L'exemple suivant supprime un analyseur existant, puis en crée un nouveau, à l'aide de l'API Python pour AWS Glue.

```
response = client.delete_crawler(
    Name='crawler_name'
)

response = client.create_crawler(
    Name='crawler_name',
    Role='IAM_role',
    DatabaseName='database_name',
    Description='string',
    Targets={
        'S3Targets': [
            {
                'Path': 'string',
                'Exclusions': [
                    'string',
                ]
            },
        ],
        'JdbcTargets': [
            {
                'ConnectionName': 'ConnectionName',
                'Path': 'Include_path',
                'Exclusions': [
                    'string',
                ]
            },
        ],
    },
    Schedule='string',
    Classifiers=[
        'string',
    ],
    TablePrefix='string',
    SchemaChangePolicy={
        'UpdateBehavior': 'LOG|UPDATE_IN_DATABASE',
    }
)
```

```

    'DeleteBehavior': 'LOG' | 'DELETE_FROM_DATABASE' | 'DEPRECATE_IN_DATABASE'
  },
  Configuration='string'
)

```

Après avoir créé l'analyseur, exécutez un analyseur qui se connecte à un ou plusieurs magasins de données, détermine les structures de données et écrit les tables dans le catalogue de données. Vous pouvez exécuter votre analyseur sur un calendrier, comme illustré ci-après.

```

response = client.start_crawler(
  Name='string'
)

```

Cet exemple utilise Amazon Redshift comme cible. Les types de données Amazon Redshift sont mappés aux types de AWS Glue données de la manière suivante après l'exécution du robot d'exploration.

Type de données Amazon Redshift	AWS Glue type de données
smallint	smallint
integer	int
bigint	bigint
decimal	decimal(18,0)
decimal(p,s)	decimal(p,s)
real	double
double precision	double
boolean	boolean
char	chaîne
varchar	chaîne
varchar(n)	chaîne

date	date
timestamp	timestamp
timestampz	timestamp

## Conversion de scripts ETL Informatica avec AWS Schema Conversion Tool

Vous pouvez utiliser l'interface de ligne de commande (CLI) de l'AWS SCT pour convertir vos scripts ETL Informatica afin de pouvoir les utiliser avec votre nouvelle base de données cible. Cette conversion comprend trois étapes clés. AWS SCT convertit d'abord le code SQL intégré à vos objets Informatica. Modifiez ensuite les noms des objets de base de données conformément aux règles de migration que vous avez spécifiées dans votre projet. Enfin, AWS SCT redirige les connexions de vos scripts ETL Informatica vers la nouvelle base de données cible.

Vous pouvez convertir des scripts ETL Informatica dans le cadre d'un projet de conversion de l'AWS SCT base de données. Assurez-vous d'ajouter vos bases de données source et cible au projet lorsque vous convertissez des scripts ETL Informatica.

Pour convertir les scripts ETL Informatica, assurez-vous d'utiliser la version 1.0.667 de l'AWS SCT ou une version ultérieure. Familiarisez-vous également avec l'interface de ligne de commande de l'AWS SCT. Pour de plus amples informations, veuillez consulter [Référence CLI pour AWS Schema Conversion Tool](#).

Pour convertir des scripts ETL Informatica à l'aide de l'AWS SCT

1. Créez un nouveau script CLI de l'AWS SCT ou modifiez un modèle de scénario existant. Par exemple, vous pouvez télécharger et modifier le `InformaticConversionTemplate.scts` modèle. Pour de plus amples informations, veuillez consulter [Obtenir des scénarios CLI](#).
2. Téléchargez les pilotes JDBC requis pour vos bases de données source et cible. Spécifiez l'emplacement de ces pilotes à l'aide de la `SetGlobalSettings` commande. Spécifiez également les dossiers dans lesquels l'AWS SCT vous pouvez enregistrer les fichiers journaux.

L'exemple de code suivant montre comment ajouter le chemin d'accès aux pilotes Oracle et PostgreSQL dans les paramètres. Après avoir exécuté cet exemple de code, l'AWS



SCT stocke les fichiers journaux dans le C:\sct\_log dossier. AWS SCT Stocke également les fichiers journaux de la console dans le C:\Temp\oracle\_postgresql dossier.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{"oracle_driver_file": "C:\\drivers\\ojdbc8.jar",
  "postgresql_driver_file": "C:\\drivers\\postgresql-42.2.19.jar" }'
/

SetGlobalSettings
  -save: 'false'
  -settings: '{
  "log_folder": "C:\\sct_log",
  "console_log_folder": "C:\\Temp\\oracle_postgresql"}'
```

3. Créez un nouveau AWS SCT projet. Entrez le nom et l'emplacement de votre projet.

L'exemple de code suivant crée le oracle\_postgresql projet dans le C:\Temp dossier.

```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. Ajoutez des informations de connexion concernant vos bases de données source et cible.

L'exemple de code suivant ajoute des bases de données Oracle et PostgreSQL en tant que source et cible pour votre projet. AWS SCT

```
AddSource
  -password: 'source_password'
  -port: '1521'
  -vendor: 'ORACLE'
  -name: 'ORACLE'
  -host: 'source_address'
  -database: 'ORCL'
  -user: 'source_user'
/

AddTarget
  -database: 'postgresql'
  -password: 'target_password'
  -port: '5432'
```

```
-vendor: 'POSTGRESQL'
-name: 'POSTGRESQL'
-host: 'target_address'
-user: 'target_user'
/
```

Dans l'exemple précédent, remplacez *source\_user* et par les noms *target\_user* des utilisateurs de votre base de données. Ensuite, remplacez *source\_password* et *target\_password* par vos mots de passe. Pour *source\_address* et *target\_address*, entrez les adresses IP de vos serveurs de base de données source et cible.

Pour vous connecter à une base de données Oracle version 19 ou ultérieure, utilisez le nom du service Oracle dans la AddSource commande. Pour ce faire, ajoutez le `-connectionType` paramètre et définissez sa valeur sur `'basic_service_name'`. Ajoutez ensuite le `-servicename` paramètre et définissez sa valeur sur le nom de votre service Oracle. Pour plus d'informations sur la AddSource commande, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

5. Créez une nouvelle règle de AWS SCT mappage qui définit les moteurs de base de données cibles pour chaque schéma de base de données source. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).

L'exemple de code suivant crée une règle de mappage qui inclut tous les schémas de base de données Oracle source et définit PostgreSQL comme cible de migration.

```
AddServerMapping
-sourceTreePath: 'Servers.ORACLE'
-targetTreePath: 'Servers.POSTGRESQL'
/
```

6. Ajoutez des informations de connexion concernant vos fichiers XML source et cible Informatica.

L'exemple de code suivant ajoute les fichiers XML Informatica à partir des `C:\Informatica_target` dossiers `C:\Informatica_source` et.

```
AddSource
-name: 'INFA_SOURCE'
-vendor: 'INFORMATICA'
-mappingsFolder: 'C:\Informatica_source'
/
AddTarget
```

```
-name: 'INFA_TARGET'  
-vendor: 'INFORMATICA'  
-mappingsFolder: 'C:\Informatica_target'  
/
```

7. Créez une autre règle de mappage pour définir le fichier XML Informatica cible pour votre fichier XML Informatica source.

L'exemple de code suivant crée une règle de mappage qui inclut les fichiers XML Informatica source et cible utilisés dans l'exemple précédent.

```
AddServerMapping  
-sourceTreePath: 'ETL.INFA_SOURCE'  
-targetTreePath: 'ETL.INFA_TARGET'  
/
```

8. Spécifiez la connexion au serveur de base de données qui correspond à la référence du nom de connexion Informatica.

L'exemple de code suivant configure la redirection de vos scripts ETL Informatica de votre source vers la nouvelle base de données cible. Cet exemple configure également les variables de connexion.

```
ConfigureInformaticaConnectionsRedirect  
-treePath: 'ETL.INFA_SOURCE.Files'  
-connections: '{  
  "ConnectionNames": [  
    {  
      "name": "Oracle_src",  
      "newName": "postgres",  
      "treePath": "Servers.ORACLE"  
    }  
  ]  
  "ConnectionVariables": [  
    {  
      "name": "$Source",  
      "treePath": "Servers.ORACLE"  
    }  
  ]  
}'  
/
```

## 9. Convertissez les schémas de votre base de données source et vos scripts ETL Informatica.

L'exemple de code suivant convertit tous vos schémas de base de données Oracle source et votre fichier XML Informatica.

```
Convert
-treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
-treePath: 'ETL.INFA_SOURCE.Files'
/
```

## 10. (Facultatif) Enregistrez votre projet de conversion et le rapport d'évaluation. Ce rapport inclut les actions de conversion et des recommandations sur la manière de les traiter.

L'exemple de code suivant enregistre votre projet et enregistre une copie du rapport d'évaluation sous forme de fichier PDF dans le C:\Temp dossier.

```
SaveProject
/
SaveReportPDF
-treePath: 'ETL.INFA_SOURCE.Files'
-file: 'C:\Temp\Informatica.pdf'
/
```

## 11. Enregistrez votre fichier XML Informatica converti.

L'exemple de code suivant enregistre le fichier XML converti dans le C:\Temp dossier. Vous avez spécifié ce dossier à l'étape précédente à l'aide de la AddTarget commande.

```
SaveTargetInformaticaXML
-treePath: 'ETL.INFA_TARGET.Files'
/
```

## 12. Enregistrez votre script sous forme de .scts fichier et exécutez-le à l'aide de la RunSCTBatch commande de la AWS SCT CLI. Pour de plus amples informations, veuillez consulter [AWS SCT Mode script CLI](#).

L'exemple suivant exécute le Informatica.scts script dans le C:\Temp dossier. Vous pouvez utiliser cet exemple sous Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Si vous modifiez vos scripts ETL Informatica source, réexécutez le script AWS SCT CLI.

## Conversion de SSIS en AWS Glue with AWS SCT

Vous trouverez ci-dessous comment convertir les packages Microsoft SQL Server Integration Services (SSIS) en packages AWS Glue Using AWS SCT.

Pour convertir des packages Microsoft SSIS AWS Glue, assurez-vous d'utiliser la AWS SCT version 1.0.642 ou supérieure. Vous devez également disposer d'un projet SSIS avec des packages ETL — .dtsx, .conmgr, et .params des fichiers dans le dossier local.

Vous n'avez pas besoin d'un serveur SSIS installé. Le processus de conversion passe par les fichiers SSIS locaux.

Pour convertir un package SSIS en AWS Glue utilisant AWS SCT

1. Créez un nouveau projet dans AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu pour ajouter un nouveau package SSIS source à votre projet.
3. Choisissez SQL Server Integration Services et effectuez les opérations suivantes :
  - Nom de la connexion — Entrez le nom de votre connexion. AWS SCT affiche ce nom dans l'arborescence des métadonnées.
  - Dossier des packages SSIS — Choisissez le chemin d'accès au dossier de votre projet SSIS contenant les packages.

AWS SCT lit les fichiers du projet (fichiers avec les extensions .dtsx, .conmgr ou .params) depuis le dossier local et les analyse. Il les organise ensuite dans une AWS SCT arborescence de catégories.

4. Choisissez Ajouter une cible dans le menu pour ajouter une nouvelle plate-forme cible afin de convertir vos packages SSIS source.
5. Choisissez AWS Glue et complétez ce qui suit :

- Nom de la connexion — Entrez le nom de votre connexion. AWS SCT affiche ce nom dans l'arborescence des métadonnées.
- Copier depuis le AWS profil — Choisissez le profil à utiliser.
- AWS clé d'accès — Entrez votre clé AWS d'accès.
- AWS clé secrète — Entrez votre clé AWS secrète.
- Région — Choisissez celle Région AWS que vous souhaitez utiliser dans la liste.
- Dossier du compartiment Amazon S3 : entrez le chemin du dossier du compartiment Amazon S3 que vous prévoyez d'utiliser.

Vous pouvez utiliser une AWS Glue cible virtuelle. Dans ce cas, il n'est pas nécessaire de spécifier les informations d'identification de connexion. Pour de plus amples informations, veuillez consulter [the section called “Cartographie de cibles virtuelles”](#).

6. Créez une nouvelle règle de mappage qui inclut votre package SSIS source et votre AWS Glue cible. Pour de plus amples informations, veuillez consulter [the section called “Nouveau mappage des types de données”](#).
7. Dans le menu Affichage, choisissez Vue principale.
8. Dans l'arborescence SSIS, ouvrez le menu contextuel (clic droit) des gestionnaires de connexions, puis choisissez Configurer les connexions.
9. Configurez le gestionnaire de connexions du projet.

Pour configurer un mappage de connexion pour les gestionnaires de connexions SSIS, spécifiez la AWS Glue connexion pour le gestionnaire de connexions SSIS correspondant. Assurez-vous que vos AWS Glue connexions sont déjà créées.

- a. Sous Connexions, sélectionnez Connexions au projet.
  - b. Pour la connexion au catalogue Glue, choisissez la AWS Glue connexion appropriée.
10. Configurez le gestionnaire de connexions aux packages :
    - a. Sous Connexions, choisissez votre forfait.
    - b. Pour la connexion au catalogue Glue, choisissez la AWS Glue connexion appropriée.
    - c. Répétez ces actions pour toutes les connexions disponibles pour votre package.
  11. Choisissez Appliquer.

12. Convertissez votre package. Dans l'arborescence des sources, recherchez Packages. Ouvrez le menu contextuel (clic droit) de votre package, puis choisissez Convertir le package.
13. Enregistrez le script converti sur Amazon S3. Dans l'arborescence cible, recherchez Package scripts. Ouvrez le menu contextuel (clic droit) de votre script converti, puis choisissez Enregistrer dans S3.
14. Configurez votre AWS Glue tâche. Dans l'arborescence cible, recherchez Package scripts. Ouvrez le menu contextuel (clic droit) de votre script converti, puis choisissez Configurer la AWS Glue tâche.
15. Complétez les trois sections de configuration :
  - a. Complétez la section relative au flux de données de conception :
    - Stratégie d'exécution : choisissez comment votre tâche exécutera les scripts ETL. Choisissez SEQUENTIAL pour exécuter les scripts dans l'ordre indiqué dans l'assistant. Choisissez PARALLEL pour exécuter les scripts en parallèle, sans tenir compte de l'ordre spécifié dans l'assistant.
    - Scripts — Choisissez le nom de votre script converti.
    - Choisissez Suivant.
  - b. Complétez la section Propriétés du job :
    - Nom — Entrez le nom de votre AWS Glue tâche.
    - Rôle IAM — Choisissez le rôle IAM qui est utilisé pour autoriser les ressources utilisées pour exécuter le travail et accéder aux magasins de données.
    - Nom du fichier de script — Entrez le nom du script converti.
    - Chemin S3 du fichier de script : entrez le chemin Amazon S3 vers votre script converti.
    - Chiffrer le script à l'aide du SSE-S3 : choisissez cette option pour protéger les données à l'aide du chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3 (SSE-S3).
    - Répertoire temporaire : entrez le chemin Amazon S3 vers un répertoire temporaire pour obtenir des résultats intermédiaires. AWS Glue et les transformations AWS Glue intégrées utilisent ce répertoire pour lire ou écrire sur Amazon Redshift.
    - AWS SCT génère automatiquement le chemin pour les bibliothèques Python. Vous pouvez consulter ce chemin dans le chemin de bibliothèque Python généré. Vous ne pouvez pas modifier ce chemin généré automatiquement. Pour utiliser des bibliothèques Python supplémentaires, entrez le chemin dans User python library path.

- Chemin de la bibliothèque Python de l'utilisateur — Entrez les chemins des bibliothèques Python utilisateur supplémentaires. Séparez les chemins Amazon S3 par des virgules.
  - Chemin des fichiers JAR dépendants : entrez les chemins des fichiers JAR dépendants. Séparez les chemins Amazon S3 par des virgules.
  - Chemin des fichiers référencés : entrez les chemins des fichiers supplémentaires, tels que les fichiers de configuration, requis par votre script. Séparez les chemins Amazon S3 par des virgules.
  - Capacité maximale — Entrez le nombre maximum d'unités de traitement de AWS Glue données (DPU) qui peuvent être allouées lors de l'exécution de cette tâche. Vous pouvez saisir un nombre entier compris entre 2 et 100. La valeur par défaut est 2.
  - Simultanéité maximale — Entrez le nombre maximum d'exécutions simultanées autorisées pour cette tâche. La valeur par défaut est 1. AWS Glue renvoie une erreur lorsque ce seuil est atteint.
  - Délai d'expiration de la tâche (minutes) : entrez la valeur du délai d'expiration de votre tâche ETL afin de vous protéger contre les tâches intempestives. La valeur par défaut est de 2 880 minutes (48 heures) pour les tâches par lots. Si la tâche dépasse cette limite, l'état d'exécution de la tâche passe à TIMEOUT.
  - Seuil de notification de retard (minutes) : entrez le seuil en minutes avant d' AWS SCT envoyer une notification de retard.
  - Nombre de tentatives : entrez le nombre de fois (0 à 10) que la tâche AWS Glue doit automatiquement redémarrer en cas d'échec. Les tâches qui atteignent le délai d'expiration ne sont pas redémarrées. La valeur par défaut est 0.
  - Choisissez Suivant.
- c. Configurez les connexions requises :
- i. Dans Toutes les connexions, choisissez les AWS Glue connexions requises et ajoutez-les à la liste des connexions sélectionnées.
  - ii. Choisissez Finish (Terminer).
16. Créez une AWS Glue tâche configurée. Dans l'arborescence cible, recherchez et développez les tâches ETL. Ouvrez le menu contextuel (clic droit) de la tâche ETL que vous avez configurée, puis choisissez Create AWS Glue Job.
17. Exécutez le AWS Glue job :
- a. Ouvrez la AWS Glue console à l'adresse <https://console.aws.amazon.com/glue/>.



- b. Dans le volet de navigation, sélectionnez Tâches.
- c. Choisissez Ajouter une tâche, puis choisissez la tâche que vous souhaitez exécuter.
- d. Dans l'onglet Actions, choisissez Run job.

## Composants SSIS AWS SCT pouvant être convertis en AWS Glue

Vous pouvez l'utiliser AWS SCT pour convertir des composants de flux de données et de flux de contrôle, ainsi que des conteneurs, des paramètres et des variables.

Les composants de flux de données pris en charge sont les suivants :

- Destination ADO NET
- Source d'ADO NET
- Regrouper
- Transformation du cache
- Transformation de la carte des personnages
- Transformation fractionnée conditionnelle
- Copier la transformation d'une colonne
- Transformation de la conversion des données
- Transformation de colonne dérivée
- Destination Excel
- Source Excel
- Transformation de colonne d'exportation
- Destination du fichier plat
- Source de fichier plat
- Transformation de Fuzzy Lookup
- Transformation de colonne d'importation
- Transformation des recherches
- Fusion, jointure et transformation
- Fusion et transformation
- Transformation en multidiffusion
- Destination ODBC

- Source ODBC
- Transformation des commandes OLE DB
- Destination OLE DB
- Source de base de données OLE
- Transformation d'échantillonnage en pourcentage
- Transformation du pivot
- Destination du fichier brut
- Source de fichier brut
- RecordSet Destination
- Transformation du nombre de lignes
- Transformation d'échantillonnage de lignes
- Transformation de tri
- Destination du serveur SQL
- Transformation totale de l'Union
- Transformation débridée
- Source XML

Les composants du flux de contrôle pris en charge sont les suivants :

- Tâche d'insertion en bloc
- Tâche d'exécution de package
- Exécuter une tâche SQL
- Exécuter la tâche d'instruction T-SQL
- Tâche d'expression
- Tâche du système de fichiers
- Notifier la tâche de l'opérateur
- Tâche d'envoi de courrier

Les conteneurs SSIS pris en charge sont les suivants :

- Pour Loop Container

- Conteneur Foreach Loop
- Conteneur de séquence

## Conversion de packages SSIS en with AWS Glue StudioAWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des packages Microsoft SQL Server Integration Services (SSIS) en AWS Glue Studio.

Un package SSIS inclut les composants nécessaires, tels que le gestionnaire de connexions, les tâches, le flux de contrôle, le flux de données, les paramètres, les gestionnaires d'événements et les variables, pour exécuter une tâche d'extraction, de transformation et de chargement (ETL) spécifique. AWS SCT convertit les packages SSIS dans un format compatible avec AWS Glue Studio. Après avoir migré votre base de données source vers le AWS Cloud, vous pouvez exécuter ces AWS Glue Studio tâches converties pour effectuer des tâches ETL.

Pour convertir des packages Microsoft SSIS AWS Glue Studio, assurez-vous d'utiliser la AWS SCT version 1.0.661 ou supérieure.

### Rubriques

- [Prérequis](#)
- [Ajouter des packages SSIS à votre AWS SCT projet](#)
- [Conversion de packages SSIS en AWS Glue Studio with AWS SCT](#)
- [Création d' AWS Glue Studio emplois à l'aide du code converti](#)
- [Création d'un rapport d'évaluation pour un package SSIS avec AWS SCT](#)
- [Composants SSIS AWS SCT pouvant être convertis en AWS Glue Studio](#)

## Prérequis

Dans cette section, découvrez les tâches préalables à la conversion des packages SSIS en AWS Glue. Ces tâches incluent la création AWS des ressources requises dans votre compte.

Vous pouvez utiliser AWS Identity and Access Management (IAM) pour définir les politiques et les rôles nécessaires pour accéder aux ressources qui AWS Glue Studio utilisent. Pour plus d'informations, consultez la section [Autorisations IAM pour l' AWS Glue Studio utilisateur](#).

Après avoir AWS SCT converti vos scripts sources en AWS Glue Studio, téléchargez les scripts convertis dans un compartiment Amazon S3. Assurez-vous de créer ce compartiment Amazon S3 et de le sélectionner dans les paramètres du profil de AWS service. Pour plus d'informations sur la création d'un compartiment S3, consultez [Créer votre premier compartiment S3](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Pour vous assurer qu'il AWS Glue Studio peut se connecter à votre banque de données, créez un connecteur personnalisé et une connexion. Stockez également les informations d'identification de la base de données dans AWS Secrets Manager.

Pour créer un connecteur personnalisé

1. Téléchargez le pilote JDBC pour votre banque de données. Pour plus d'informations sur les pilotes JDBC AWS SCT utilisés, consultez. [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#)
2. Téléchargez ce fichier de pilote dans votre compartiment Amazon S3. Pour plus d'informations, consultez la section [Charger un objet dans votre compartiment](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.
3. Connectez-vous à la AWS Glue Studio console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/gluestudio/>.
4. Choisissez Connecteurs, puis sélectionnez Créer un connecteur personnalisé.
5. Pour l'URL du connecteur S3, choisissez Browse S3, puis le fichier de pilote JDBC que vous avez chargé dans votre compartiment Amazon S3.
6. Entrez un nom descriptif pour votre connecteur. Par exemple, saisissez **SQLServer**.
7. Pour le type de connecteur, choisissez JDBC.
8. Dans le champ Nom de la classe, entrez le nom de la classe principale de votre pilote JDBC. Pour SQL Server, entrez **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
9. Pour la base d'URL JDBC, entrez l'URL de base JDBC. La syntaxe de l'URL de base JDBC dépend du moteur de base de données source. Pour SQL Server, utilisez le format suivant : **jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>**  
  
Assurez-vous de remplacer **<host>**, **<port>**, **<dbname>**, **<username>**, et **<password>** par vos valeurs.
10. Pour le séparateur de paramètres d'URL, entrez le point-virgule (.) ;
11. Sélectionnez Créer un connecteur.

## Pour stocker les informations d'identification de base de données dans AWS Secrets Manager

1. Connectez-vous à la AWS Secrets Manager console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/secretsmanager/>.
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. Sur la page Choose secret type (Choisir un type de secret), procédez comme suit :
  - a. Pour Type de secret, choisissez l'autre type de secret.
  - b. Pour les paires clé/valeur, entrez les clés suivantes : **host**, **port** **dbnameusername**, et **password**  
  
Entrez ensuite les valeurs de ces clés.
4. Sur la page Configurer le secret, entrez un nom de secret descriptif. Par exemple, saisissez **SQL\_Server\_secret**.
5. Choisissez Suivant. Ensuite, sur la page Configurer la rotation, sélectionnez à nouveau Suivant.
6. Dans la page Review (Révision), passez en revue vos paramètres, puis choisissez Store (Stocker).

## Pour créer une connexion pour votre connecteur

1. Connectez-vous à la AWS Glue Studio console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/gluestudio/>.
2. Choisissez le connecteur pour lequel vous souhaitez créer une connexion, puis choisissez Créer une connexion.
3. Sur la page Créer une connexion, entrez un nom descriptif pour votre connexion. Par exemple, saisissez **SQL-Server-connection**.
4. Dans AWS Secret, choisissez le secret que vous avez créé dans AWS Secrets Manager.
5. Configurez les options réseau, puis choisissez Créer une connexion.

Vous pouvez désormais créer une AWS Glue Studio tâche à l'aide d'un connecteur personnalisé. Pour de plus amples informations, veuillez consulter [Création d' AWS Glue Studio emplois](#).

## Ajouter des packages SSIS à votre AWS SCT projet

Vous pouvez ajouter plusieurs packages SSIS à un même AWS SCT projet.

## Pour ajouter un package SSIS à votre AWS SCT projet

1. Créez un nouveau projet avec AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu, puis sélectionnez SQL Server Integration Services.
3. Dans Nom de connexion, entrez le nom de vos packages SSIS. AWS SCT affiche ce nom dans l'arborescence du panneau de gauche.
4. Pour le dossier des packages SSIS, entrez le chemin d'accès au dossier contenant les packages SSIS source.
5. Choisissez Ajouter une cible dans le menu, puis choisissez AWS Glue Studio.

Pour vous connecter à AWS Glue Studio, AWS SCT utilise votre AWS profil. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).

6. Créez une règle de mappage, qui inclut votre package SSIS source et votre AWS Glue Studio cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).
7. Créez AWS Glue Studio des connexions dans la AWS Glue Studio console. Pour plus d'informations, consultez la section [Création de connexions pour les connecteurs](#).
8. Choisissez Gestionnaires de connexions dans l'arborescence de gauche, ouvrez le menu contextuel (clic droit), puis sélectionnez Configurer les connexions.

AWS SCT affiche la fenêtre Configurer les connexions.

9. Pour chaque connexion SSIS source, choisissez une AWS Glue Studio connexion.

## Conversion de packages SSIS en AWS Glue Studio with AWS SCT

Ci-dessous, découvrez comment convertir les packages SSIS à AWS Glue Studio l'utilisation AWS SCT.

### Pour convertir un package SSIS en AWS Glue Studio

1. Ajoutez votre package SSIS à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des packages SSIS à votre AWS SCT projet](#).
2. Dans le panneau de gauche, développez les nœuds ETL et SSIS.

3. Choisissez Packages, ouvrez le menu contextuel (clic droit), puis choisissez Convertir le package.

AWS SCT convertit les packages SSIS sélectionnés en fichiers JSON. Ces objets JSON représentent un nœud dans un graphe acyclique dirigé (DAG). Trouvez vos fichiers convertis dans le DAGs nœud Package dans l'arborescence de droite.

4. Choisissez Package DAGs, ouvrez le menu contextuel (clic droit), puis sélectionnez Enregistrer sur Amazon S3.

Vous pouvez désormais utiliser ces scripts pour créer des tâches dans le AWS Glue Studio.

## Création d' AWS Glue Studio emplois à l'aide du code converti

Après avoir converti vos packages SSIS source, vous pouvez utiliser les fichiers JSON convertis pour créer des AWS Glue Studio tâches.

Pour créer un AWS Glue Studio emploi

1. Choisissez Package DAGs dans l'arborescence de droite, ouvrez le menu contextuel (clic droit), puis choisissez Configurer la AWS Glue Studio tâche.
2. (Facultatif) Appliquez le pack d'extension qui émule les fonctions SSIS dans AWS Glue Studio.
3. La fenêtre Configurer la AWS Glue Studio tâche s'ouvre.

Complétez la section Propriétés de base de la tâche :

- Nom — Entrez le nom de votre AWS Glue Studio travail.
- Nom du fichier de script — Entrez le nom de votre script de tâche.
- Paramètres du job : ajoutez des paramètres et entrez leurs valeurs.

Choisissez Suivant.

4. Complétez la section Propriétés avancées de la tâche :
  - Rôle IAM — Choisissez le rôle IAM utilisé pour l'autorisation AWS Glue Studio et l'accès aux magasins de données.
  - Chemin S3 du fichier de script : entrez le chemin Amazon S3 vers votre script converti.

- Répertoire temporaire : entrez le chemin Amazon S3 vers un répertoire temporaire pour obtenir des résultats intermédiaires. AWS Glue Studio utilise ce répertoire pour lire ou écrire sur Amazon Redshift.
- AWS SCT génère automatiquement le chemin pour les bibliothèques Python. Vous pouvez consulter ce chemin dans le chemin de bibliothèque Python généré. Vous ne pouvez pas modifier ce chemin généré automatiquement. Pour utiliser des bibliothèques Python supplémentaires, entrez le chemin dans User python library path.
- Chemin de la bibliothèque Python de l'utilisateur — Entrez les chemins des bibliothèques Python utilisateur supplémentaires. Séparez les chemins Amazon S3 par des virgules.
- Chemin des fichiers JAR dépendants : entrez les chemins des \*.jar fichiers dépendants. Séparez les chemins Amazon S3 par des virgules.
- Chemin des fichiers référencés : entrez les chemins des fichiers supplémentaires, tels que les fichiers de configuration, requis par votre script. Séparez les chemins Amazon S3 par des virgules.
- Type de travailleur — Choisissez G.1X ou G.2X.

Lorsque vous choisissez, G.1X chaque worker est mappé sur 1 DPU (4 vCPU, 16 Go de mémoire et 64 Go de disque).

Lorsque vous choisissez, G.2X chaque worker est mappé sur 2 DPU (8 vCPU, 32 Go de mémoire et 128 Go de disque).

- Nombre de travailleurs demandé : entrez le nombre de travailleurs alloués lors de l'exécution de la tâche.
- Simultanéité maximale — Entrez le nombre maximum d'exécutions simultanées autorisées pour cette tâche. La valeur par défaut est 1. AWS Glue renvoie une erreur lorsque ce seuil est atteint.
- Délai d'expiration de la tâche (minutes) : entrez la valeur du délai d'expiration de votre tâche ETL afin de vous protéger contre les tâches intempestives. La valeur par défaut est de 2 880 minutes (48 heures) pour les tâches par lots. Si la tâche dépasse cette limite, l'état d'exécution de la tâche passe à TIMEOUT.
- Seuil de notification de retard (minutes) : entrez le seuil en minutes avant d' AWS SCT envoyer une notification de retard.
- Nombre de tentatives : entrez le nombre de fois (0 à 10) que la tâche AWS Glue doit automatiquement redémarrer en cas d'échec. Les tâches qui atteignent le délai d'expiration ne sont pas redémarrées. La valeur par défaut est 0.



Choisissez Finish (Terminer).

AWS SCT configure les AWS Glue Studio tâches que vous avez sélectionnées.

5. Trouvez vos tâches configurées sous Tâches ETL dans l'arborescence appropriée. Choisissez votre tâche configurée, ouvrez le menu contextuel (clic droit), puis choisissez Créer une AWS Glue Studio tâche.
6. Choisissez le statut Appliquer et assurez-vous que la valeur du statut de votre offre d'emploi est Réussite.
7. Ouvrez la AWS Glue Studio console, choisissez Actualiser, puis choisissez votre tâche. Choisissez ensuite Next (Suivant).

## Création d'un rapport d'évaluation pour un package SSIS avec AWS SCT

Le rapport d'évaluation de la migration ETL fournit des informations sur la conversion de vos packages SSIS dans un format compatible avec AWS Glue Studio. Le rapport d'évaluation inclut des mesures à prendre pour les composants de vos packages SSIS. Ces actions indiquent quels composants ne AWS SCT peuvent pas être convertis automatiquement.

Pour créer un rapport d'évaluation de la migration ETL

1. Développez le nœud SSIS sous ETL dans le panneau de gauche.
2. Choisissez Packages, ouvrez le menu contextuel (clic droit), puis choisissez Créer un rapport.
3. Consultez l'onglet Résumé. AWS SCT Affiche ici les informations récapitulatives du rapport d'évaluation de la migration ETL. Il inclut les résultats de conversion pour tous les composants de vos packages SSIS.
4. (Facultatif) Enregistrez une copie locale du rapport d'évaluation de la migration ETL sous forme de fichier PDF ou de fichier CSV (valeurs séparées par des virgules) :
  - Pour enregistrer le rapport d'évaluation de la migration ETL sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.

Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations pour la conversion des scripts.

- Pour enregistrer le rapport d'évaluation de la migration ETL sous forme de fichier CSV, choisissez Enregistrer au format CSV en haut à droite.

AWS SCT crée trois fichiers CSV. Ces fichiers contiennent des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.

5. Choisissez l'onglet **Éléments d'action**. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle en AWS Glue Studio. Lorsque vous choisissez un élément d'action dans la liste, AWS SCT met en surbrillance l'élément de votre package SSIS source auquel l'action s'applique.

## Composants SSIS AWS SCT pouvant être convertis en AWS Glue Studio

Vous pouvez l'utiliser AWS SCT pour convertir les composants et paramètres du flux de données SSIS en AWS Glue Studio.

Les composants de flux de données pris en charge sont les suivants :

- Destination ADO NET
- Source d'ADO NET
- Regrouper
- Carte des personnages
- Fractionnement conditionnel
- Copier la colonne
- Conversion des données
- Colonne dérivée
- Chercher
- Fusionner
- Joindre par fusion
- Multicast
- ODBC Destination
- ODBC Source
- OLEDB Destination
- OLEDB Source
- Nombre de lignes
- Tri

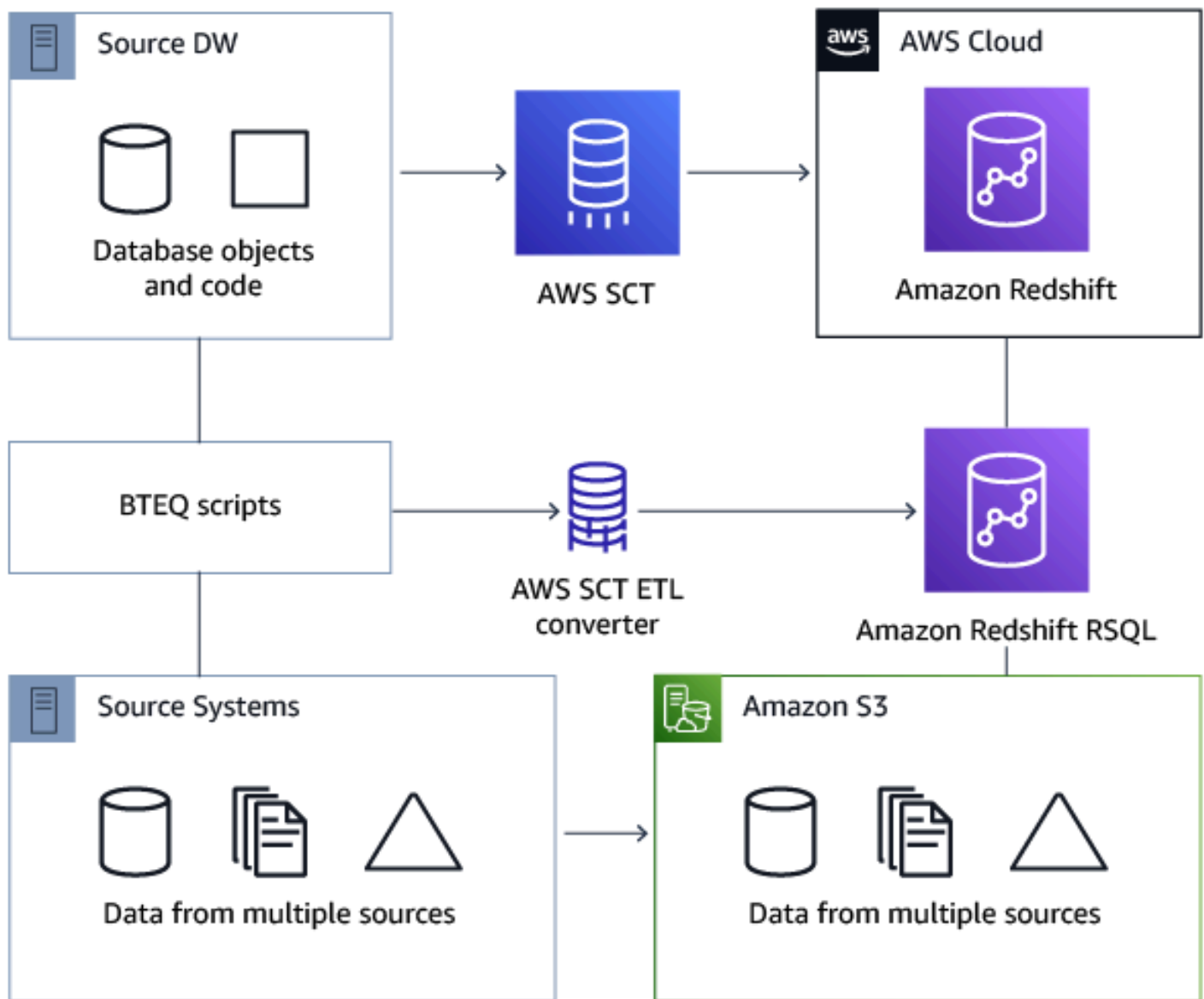
- Destination du serveur SQL
- Tout syndiquer

AWS SCT peut convertir davantage de composants SSIS en AWS Glue. Pour de plus amples informations, veuillez consulter [Composants SSIS AWS SCT pouvant être convertis en AWS Glue](#).

## Conversion de scripts Teradata BTEQ en Amazon Redshift RSQL avec AWS SCT

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir les scripts Teradata Basic Teradata Query (BTEQ) en Amazon Redshift RSQL.

Le schéma d'architecture suivant montre le projet de migration de base de données qui inclut la conversion de scripts d'extraction, de transformation et de chargement (ETL) en Amazon Redshift RSQL.



## Rubriques

- [Ajouter des scripts BTEQ à votre projet AWS SCT](#)
- [Configuration des variables de substitution dans les scripts BTEQ avec AWS SCT](#)
- [Conversion de scripts Teradata BTEQ en Amazon Redshift RSQL avec AWS SCT](#)
- [Gérer les scripts BTEQ avec AWS SCT](#)
- [Création d'un rapport d'évaluation de la conversion des scripts BTEQ avec AWS SCT](#)
- [Modifier et enregistrer vos scripts BTEQ convertis avec AWS SCT](#)

## Ajouter des scripts BTEQ à votre projet AWS SCT

Vous pouvez ajouter plusieurs scripts à un même AWS SCT projet.

Pour ajouter un script BTEQ à votre projet AWS SCT

1. Créez un nouveau projet dans AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu, puis Teradata pour ajouter votre base de données source au projet. Pour de plus amples informations, veuillez consulter [Bases de données Teradata](#).
3. Choisissez Ajouter une cible dans le menu pour ajouter une base de données Amazon Redshift cible à votre AWS SCT projet.

Vous pouvez utiliser une plate-forme de base de données cible Amazon Redshift virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Créez une nouvelle règle de mappage qui inclut votre base de données Teradata source et votre cible Amazon Redshift. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
5. Dans le menu Affichage, choisissez Vue principale.
6. Dans le panneau de gauche, développez le nœud Scripts.
7. Choisissez des scripts BTEQ, ouvrez le menu contextuel (clic droit), puis choisissez Charger des scripts.
8. Entrez l'emplacement du code source de vos scripts Teradata BTEQ et choisissez Select folder.

AWS SCT affiche la fenêtre Charger des scripts.

9. Effectuez l'une des actions suivantes :
  - a. Si vos scripts Teradata BTEQ n'incluent pas les variables de substitution, choisissez Aucune variable de substitution, puis cliquez sur OK pour ajouter des scripts à votre projet. AWS SCT
  - b. Si vos scripts Teradata BTEQ incluent des variables de substitution, configurez-les. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts BTEQ](#).

# Configuration des variables de substitution dans les scripts BTEQ avec AWS SCT

Vos scripts Teradata BTEQ peuvent inclure des variables de substitution. Par exemple, vous pouvez utiliser un script BTEQ avec des variables de substitution pour exécuter le même ensemble de commandes sur plusieurs environnements de base de données. Vous pouvez l'utiliser AWS SCT pour configurer des variables de substitution dans vos scripts BTEQ.

Avant d'exécuter un script BTEQ avec des variables de substitution, assurez-vous d'attribuer les valeurs à toutes les variables. Pour ce faire, vous pouvez utiliser d'autres outils ou applications tels qu'un script Bash, UC4 (Automic), etc. AWS SCT ne peut résoudre et convertir des variables de substitution qu'après avoir attribué leurs valeurs.

Pour configurer les variables de substitution dans votre script BTEQ

1. Ajoutez vos scripts BTEQ à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts BTEQ à votre projet AWS SCT](#).

Lorsque vous ajoutez vos scripts, sélectionnez Les variables de substitution sont utilisées.

2. Pour Définir le format de variable, entrez une expression régulière qui correspond à toutes les variables de substitution de votre script.

Par exemple, si les noms de vos variables de substitution commencent par `#{` et se terminent par `}`, utilisez l'expression `\#{\w+}` régulière. Pour faire correspondre les variables de substitution qui commencent par le signe dollar ou par le signe pourcentage, utilisez l'expression `[\$|%\w+]` régulière.

Expressions régulières AWS SCT conformes à la syntaxe des expressions régulières Java. Pour plus d'informations, consultez le [modèle de classe java.util.regex](#) dans la documentation Java.

3. Cliquez sur OK pour charger des scripts dans votre AWS SCT projet, puis sur OK pour fermer la fenêtre Charger des scripts.
4. Choisissez Variables pour afficher toutes les variables de substitution découvertes et leurs valeurs.
5. Dans Valeur, entrez la valeur de la variable de substitution.

# Conversion de scripts Teradata BTEQ en Amazon Redshift RSQL avec AWS SCT

Découvrez ci-dessous comment convertir des scripts ETL BTEQ en Amazon Redshift RSQL à l'aide de AWS SCT

Pour convertir un script Teradata BTEQ en Amazon Redshift RSQL

1. Ajoutez vos scripts BTEQ à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts BTEQ à votre projet AWS SCT](#).
2. Configurez les variables de substitution. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts BTEQ](#).
3. Dans le panneau de gauche, développez le nœud Scripts.
4. Effectuez l'une des actions suivantes :
  - Pour convertir un seul script BTEQ, développez le nœud des scripts BTEQ, choisissez le script à convertir, puis choisissez Convertir en RSQL dans le menu contextuel (clic droit).
  - Pour convertir plusieurs scripts, assurez-vous de sélectionner tous les scripts à convertir. Choisissez ensuite les scripts BTEQ, ouvrez le menu contextuel (clic droit), puis choisissez Convertir en RSQL sous Convertir le script.

AWS SCT convertit tous les scripts Teradata BTEQ sélectionnés dans un format compatible avec Amazon Redshift RSQL. Trouvez vos scripts convertis dans le nœud Scripts du panneau de base de données cible.

5. Modifiez vos scripts Amazon Redshift RSQL convertis ou enregistrez-les. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de vos scripts BTEQ convertis](#).

## Gérer les scripts BTEQ avec AWS SCT

Vous pouvez ajouter plusieurs scripts BTEQ ou supprimer un script BTEQ de votre projet. AWS SCT

Pour ajouter un script BTEQ supplémentaire à votre projet AWS SCT

1. Développez le nœud Scripts dans le panneau de gauche.
2. Choisissez le nœud des scripts BTEQ et ouvrez le menu contextuel (clic droit).
3. Choisissez Charger des scripts.

4. Entrez les informations requises pour ajouter un nouveau script BTEQ et configurer les variables de substitution. Pour plus d'informations, consultez [Ajouter des scripts BTEQ à votre projet AWS SCT](#) et [Configuration des variables de substitution dans les scripts BTEQ](#).

Pour supprimer un script BTEQ de votre projet AWS SCT

1. Développez le nœud de scripts BTEQ sous Scripts dans le panneau de gauche.
2. Choisissez le script à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer le script.

## Création d'un rapport d'évaluation de la conversion des scripts BTEQ avec AWS SCT

Un rapport d'évaluation de la conversion des scripts BTEQ fournit des informations sur la conversion des commandes BTEQ et des instructions SQL de vos scripts BTEQ dans un format compatible avec Amazon Redshift RSQL. Le rapport d'évaluation inclut des actions pour les commandes BTEQ et les instructions SQL qui ne AWS SCT peuvent pas être converties.

Pour créer un rapport d'évaluation de la conversion des scripts BTEQ

1. Développez le nœud de scripts BTEQ sous Scripts dans le panneau de gauche.
2. Choisissez le script à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Conversion en RSQL sous Créer un rapport.
4. Consultez l'onglet Résumé. L'onglet Résumé affiche les informations de synthèse du rapport d'évaluation du script BTEQ. Il inclut les résultats de conversion pour toutes les commandes BTEQ et les instructions SQL de vos scripts BTEQ.
5. (Facultatif) Enregistrez une copie locale du rapport d'évaluation de la conversion des scripts BTEQ sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :

- Pour enregistrer le rapport d'évaluation de la conversion du script BTEQ sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.

Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations pour la conversion des scripts.

- Pour enregistrer le rapport d'évaluation de la conversion du script BTEQ sous forme de fichier CSV, choisissez Enregistrer au format CSV en haut à droite.



Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.

6. Choisissez l'onglet **Éléments d'action**. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle vers Amazon Redshift RSQL. Lorsque vous choisissez une action dans la liste, mettez AWS SCT en surbrillance l'élément de votre script BTEQ source auquel l'action s'applique.

## Modifier et enregistrer vos scripts BTEQ convertis avec AWS SCT

Vous pouvez modifier vos scripts convertis dans le panneau inférieur de votre AWS SCT projet. AWS SCT enregistre le script modifié dans le cadre de votre projet.

Pour enregistrer vos scripts convertis

1. Développez le nœud de scripts RSQL sous Scripts dans le panneau de base de données cible.
2. Choisissez votre script converti, ouvrez le menu contextuel (clic droit) et choisissez Enregistrer le script.
3. Entrez le chemin d'accès au dossier pour enregistrer le script converti et choisissez Enregistrer.

AWS SCT enregistre le script converti dans un fichier et ouvre ce fichier.

## Conversion de scripts shell contenant des commandes BTEQ intégrées en Amazon Redshift RSQL avec AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir des scripts shell contenant des commandes Teradata Basic Teradata Query (BTEQ) intégrées en scripts shell contenant des commandes Amazon Redshift RSQL intégrées.

AWS SCT extrait les commandes Teradata BTEQ de vos scripts shell et les convertit dans un format compatible avec Amazon Redshift. Après avoir migré la base de données Teradata vers Amazon Redshift, vous pouvez utiliser ces scripts convertis pour gérer votre nouvelle base de données Amazon Redshift.

Vous pouvez également les utiliser AWS SCT pour convertir des fichiers contenant des scripts Teradata BTEQ ETL en Amazon Redshift RSQL. Pour de plus amples informations, veuillez consulter [Conversion de scripts Teradata BTEQ en Amazon Redshift RSQL avec AWS SCT](#).

## Rubriques

- [Ajout de scripts shell avec des commandes Teradata BTEQ intégrées à votre projet AWS SCT](#)
- [Configuration de variables de substitution dans des scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT](#)
- [Conversion de scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT](#)
- [Gestion de scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT](#)
- [Création d'un rapport d'évaluation pour la conversion d'un script shell avec AWS SCT](#)
- [Modifier et enregistrer vos scripts shell convertis avec AWS SCT](#)

## Ajout de scripts shell avec des commandes Teradata BTEQ intégrées à votre projet AWS SCT

Vous pouvez ajouter plusieurs scripts à un même AWS SCT projet.

Pour ajouter un script shell à votre AWS SCT projet

1. Créez un nouveau projet dans AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu, puis Teradata pour ajouter votre base de données source au projet. Pour de plus amples informations, veuillez consulter [Bases de données Teradata](#).
3. Choisissez Ajouter une cible dans le menu pour ajouter une base de données Amazon Redshift cible à votre AWS SCT projet.

Vous pouvez utiliser une plate-forme de base de données cible Amazon Redshift virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Créez une nouvelle règle de mappage qui inclut votre base de données Teradata source et votre cible Amazon Redshift. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
5. Dans le menu Affichage, choisissez Vue principale.

6. Dans le panneau de gauche, développez le nœud Scripts.
7. Choisissez Shell, ouvrez le menu contextuel (clic droit), puis choisissez Charger des scripts.
8. Entrez l'emplacement de vos scripts shell source avec les commandes Teradata BTEQ intégrées et choisissez Select folder.

AWS SCT affiche la fenêtre Charger des scripts.

9. Effectuez l'une des actions suivantes :
  - Si vos scripts shell n'incluent pas les variables de substitution, choisissez Aucune variable de substitution, puis cliquez sur OK pour ajouter des scripts à votre AWS SCT projet.
  - Si vos scripts shell incluent des variables de substitution, configurez-les. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts shell](#).

## Configuration de variables de substitution dans des scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT

Vos scripts shell peuvent inclure des variables de substitution. Par exemple, vous pouvez utiliser un seul script avec des variables de substitution pour gérer des bases de données dans différents environnements. Vous pouvez l'utiliser AWS SCT pour configurer des variables de substitution dans vos scripts shell.

Avant d'exécuter des commandes BTEQ avec des variables de substitution à partir d'un script shell, assurez-vous d'attribuer les valeurs à toutes les variables de ce script shell. AWS SCT ne peut résoudre et convertir des variables de substitution qu'après avoir attribué leurs valeurs.

Pour configurer des variables de substitution dans votre script shell

1. Ajoutez vos scripts shell source à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts shell à votre AWS SCT projet](#).

Lorsque vous ajoutez vos scripts, sélectionnez Les variables de substitution sont utilisées.

2. Pour Définir le format de variable, entrez une expression régulière qui correspond à toutes les variables de substitution de votre script.

Par exemple, si les noms de vos variables de substitution commencent par `{` et se terminent par `}`, utilisez l'expression `\${\w+}` régulière. Pour faire correspondre les variables de

substitution qui commencent par le signe dollar ou par le signe pourcentage, utilisez l'expression `\$\w+|\%\w+` régulière.

Expressions régulières AWS SCT conformes à la syntaxe des expressions régulières Java. Pour plus d'informations, consultez le [modèle de classe java.util.regex dans](#) la documentation Java.

3. Cliquez sur OK pour charger des scripts dans votre AWS SCT projet, puis sur OK pour fermer la fenêtre Charger des scripts.
4. Choisissez Variables pour afficher toutes les variables de substitution découvertes et leurs valeurs.
5. Dans Valeur, entrez la valeur de la variable de substitution.

## Conversion de scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT

Découvrez ci-dessous comment convertir des scripts shell contenant des commandes Teradata BTEQ intégrées en scripts shell contenant des commandes Amazon Redshift RSQL intégrées à l'aide de AWS SCT

Pour convertir un script shell

1. Ajoutez vos scripts shell à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts shell à votre AWS SCT projet](#).
2. Configurez les variables de substitution. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts shell](#).
3. Dans le panneau de gauche, développez le nœud Scripts.
4. Effectuez l'une des actions suivantes :
  - Pour convertir des commandes BTEQ à partir d'un seul script shell, développez le nœud Shell, choisissez le script à convertir, puis choisissez Convertir le script dans le menu contextuel (clic droit).
  - Pour convertir plusieurs scripts, assurez-vous de sélectionner tous les scripts à convertir. Choisissez ensuite Shell, ouvrez le menu contextuel (clic droit), puis choisissez Convertir le script.
5. Choisissez OK.

AWS SCT convertit les commandes BTEQ des scripts shell que vous avez sélectionnés dans un format compatible avec Amazon Redshift RSQL. Trouvez vos scripts convertis dans le nœud Scripts du panneau de base de données cible.

6. Modifiez vos scripts Amazon Redshift RSQL convertis ou enregistrez-les. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de vos scripts shell convertis](#).

## Gestion de scripts shell avec des commandes Teradata BTEQ intégrées avec AWS SCT

Vous pouvez ajouter plusieurs scripts shell ou supprimer un script shell de votre AWS SCT projet.

Pour ajouter un nouveau script shell à votre AWS SCT projet

1. Développez le nœud Scripts dans le panneau de gauche.
2. Choisissez le nœud Shell et ouvrez le menu contextuel (clic droit).
3. Choisissez Charger des scripts.
4. Entrez les informations requises pour ajouter un nouveau script shell et configurer les variables de substitution. Pour plus d'informations, consultez [Ajouter des scripts shell à votre AWS SCT projet](#) et [Configuration des variables de substitution dans les scripts shell](#).

Pour supprimer un script shell de votre AWS SCT projet

1. Développez le nœud Shell sous Scripts dans le panneau de gauche.
2. Choisissez le script à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer le script.

## Création d'un rapport d'évaluation pour la conversion d'un script shell avec AWS SCT

Le rapport d'évaluation de la conversion des scripts shell fournit des informations sur la conversion des commandes BTEQ et des instructions SQL. La conversion s'effectue à partir de vos scripts source vers un format compatible avec Amazon Redshift RSQL. Le rapport d'évaluation inclut des actions pour les commandes BTEQ et les instructions SQL qui ne AWS SCT peuvent pas être converties.

## Pour créer un rapport d'évaluation de la conversion des scripts shell

1. Développez le nœud Shell sous Scripts dans le panneau de gauche.
2. Choisissez le script à convertir, ouvrez le menu contextuel (clic droit), puis choisissez Créer un rapport.
3. Consultez l'onglet Résumé. L'onglet Résumé affiche les informations de synthèse issues du rapport d'évaluation du script shell. Il inclut les résultats de conversion pour toutes les commandes BTEQ et instructions SQL de vos scripts source.
4. (Facultatif) Enregistrez une copie locale du rapport d'évaluation de la conversion du script shell sous forme de fichier PDF ou de fichier CSV (valeurs séparées par des virgules) :
  - Pour enregistrer le rapport d'évaluation de la conversion du script shell sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.

Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations pour la conversion des scripts.
  - Pour enregistrer le rapport d'évaluation de la conversion du script shell sous forme de fichier CSV, choisissez Enregistrer au format CSV en haut à droite.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.
5. Choisissez l'onglet Éléments d'action. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle vers Amazon Redshift RSQL. Lorsque vous sélectionnez un élément d'action dans la liste AWS SCT , surligne l'élément de votre script shell source auquel l'action s'applique.

## Modifier et enregistrer vos scripts shell convertis avec AWS SCT

Vous pouvez modifier vos scripts convertis dans le panneau inférieur de votre AWS SCT projet. AWS SCT enregistre le script modifié dans le cadre de votre projet.

### Pour enregistrer vos scripts convertis

1. Développez le nœud de scripts RSQL sous Scripts dans le panneau de base de données cible.
2. Choisissez votre script converti, ouvrez le menu contextuel (clic droit) et choisissez Enregistrer le script.
3. Entrez le chemin d'accès au dossier pour enregistrer le script converti et choisissez Enregistrer.

AWS SCT enregistre le script converti dans un fichier et ouvre ce fichier.

## Conversion de FastExport scripts vers Amazon Redshift RSQL avec AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir les scripts de FastExport travail Teradata en Amazon Redshift RSQL.

Un script de FastExport tâche est un ensemble de FastExport commandes et d'instructions SQL qui sélectionnent et exportent des données depuis la base de données Teradata. AWS SCT convertit FastExport les commandes et les instructions SQL dans un format compatible avec Amazon Redshift RSQL. Après avoir migré la base de données Teradata vers Amazon Redshift, vous pouvez utiliser ces scripts convertis pour exporter des données depuis la base de données Amazon Redshift.

### Rubriques

- [Ajouter des scripts de FastExport travail à votre AWS SCT projet](#)
- [Configuration des variables de substitution dans les scripts de FastExport travail Teradata avec AWS SCT](#)
- [Conversion de scripts de FastExport travail Teradata avec AWS SCT](#)
- [Gestion des scripts de FastExport travail Teradata avec AWS SCT](#)
- [Création d'un rapport d'évaluation pour la conversion d'un script de FastExport travail Teradata avec AWS SCT](#)
- [Modification et enregistrement de vos scripts de FastExport travail Teradata convertis avec AWS SCT](#)

## Ajouter des scripts de FastExport travail à votre AWS SCT projet

Vous pouvez ajouter plusieurs scripts à un même AWS SCT projet.

Pour ajouter un script de FastExport tâche à votre AWS SCT projet

1. Créez un nouveau projet AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called "Démarrage et gestion de projets"](#).

2. Choisissez Ajouter une source dans le menu, puis Teradata pour ajouter votre base de données source au projet. Pour de plus amples informations, veuillez consulter [Bases de données Teradata](#).
3. Choisissez Ajouter une cible dans le menu pour ajouter une base de données Amazon Redshift cible à votre AWS SCT projet.

Vous pouvez utiliser une plate-forme de base de données cible Amazon Redshift virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Créez une nouvelle règle de mappage qui inclut votre base de données Teradata source et votre cible Amazon Redshift. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
5. Dans le menu Affichage, choisissez Vue principale.
6. Dans le panneau de gauche, développez le nœud Scripts.
7. Choisissez FastExport, ouvrez le menu contextuel (clic droit), puis choisissez Charger des scripts.
8. Entrez l'emplacement du code source de vos scripts de FastExport travail Teradata et choisissez Select folder.

AWS SCT affiche la fenêtre Charger des scripts.

9. Effectuez l'une des actions suivantes :
  - Si vos scripts de FastExport travail Teradata n'incluent pas les variables de substitution, choisissez Aucune variable de substitution, puis cliquez sur OK pour ajouter des scripts à votre AWS SCT projet.
  - Si vos scripts de FastExport travail Teradata incluent des variables de substitution, configurez-les. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de FastExport travail](#).

## Configuration des variables de substitution dans les scripts de FastExport travail Teradata avec AWS SCT

Vos scripts de FastExport travail Teradata peuvent inclure des variables de substitution. Par exemple, vous pouvez utiliser un seul script avec des variables de substitution pour exporter les données



de plusieurs bases de données. Vous pouvez l'utiliser AWS SCT pour configurer des variables de substitution dans vos scripts Teradata.

Avant d'exécuter un script de FastExport travail avec des variables de substitution, assurez-vous d'attribuer les valeurs à toutes les variables. Pour ce faire, vous pouvez utiliser d'autres outils ou applications tels qu'un script Bash, UC4 (Automic), etc. AWS SCT ne peut résoudre et convertir des variables de substitution qu'après avoir attribué leurs valeurs.

Pour configurer les variables de substitution dans votre script de FastExport tâche

1. Ajoutez vos scripts de FastExport travail Teradata sources à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts BTEQ à votre projet AWS SCT](#) .

Lorsque vous ajoutez vos scripts, sélectionnez Les variables de substitution sont utilisées.

2. Pour Définir le format de variable, entrez une expression régulière correspondant à toutes les variables de substitution de votre script.

Par exemple, si les noms de vos variables de substitution commencent par `{` et se terminent par `}`, utilisez l'expression `\${\w+}` régulière. Pour faire correspondre les variables de substitution qui commencent par le signe dollar ou par le signe pourcentage, utilisez l'expression `\$\w+|\%\w+` régulière.

Expressions régulières AWS SCT conformes à la syntaxe des expressions régulières Java. Pour plus d'informations, consultez le [modèle de classe java.util.regex](#) dans la documentation Java.

3. Cliquez sur OK pour charger des scripts dans votre AWS SCT projet, puis sur OK pour fermer la fenêtre Charger des scripts.
4. Dans le panneau de gauche, développez le nœud Scripts. Choisissez FastExport, puis choisissez votre dossier contenant les scripts. Ouvrez le menu contextuel (clic droit), puis choisissez Exporter les variables sous Variables de substitution.
5. Exportez les variables de substitution pour un script. Complétez votre dossier avec des scripts, choisissez votre script, ouvrez le menu contextuel (clic droit) et choisissez Exporter les variables sous Variables de substitution.
6. Entrez le nom du fichier de valeurs séparées par des virgules (CSV) pour enregistrer les variables de substitution et choisissez Enregistrer.
7. Ouvrez ce fichier CSV et renseignez les valeurs des variables de substitution.

En fonction du système d'exploitation, AWS SCT utilise différents formats pour les fichiers CSV. Les valeurs du fichier peuvent être placées entre guillemets ou non. Assurez-vous d'utiliser

le même format pour les valeurs des variables de substitution que pour les autres valeurs du fichier. AWS SCT Impossible d'importer le fichier CSV contenant des valeurs dans différents formats.

8. Enregistrez le fichier CSV.
9. Dans le panneau de gauche, développez le nœud Scripts. Choisissez FastExport, puis choisissez votre script. Ouvrez le menu contextuel (clic droit), puis choisissez Importer des variables sous Variables de substitution.
10. Choisissez votre fichier CSV, puis sélectionnez Ouvrir.
11. Choisissez Variables pour afficher toutes les variables de substitution découvertes et leurs valeurs.

## Conversion de scripts de FastExport travail Teradata avec AWS SCT

Découvrez ci-dessous comment convertir une FastExport tâche Teradata en Amazon Redshift RSQL à l'aide de. AWS SCT

Pour convertir un script de FastExport tâche Teradata en Amazon Redshift RSQL

1. Ajoutez vos scripts de FastExport travail à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts de FastExport travail à votre AWS SCT projet](#).
2. Configurez les variables de substitution. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de FastExport travail](#).
3. Dans le panneau de gauche, développez le nœud Scripts.
4. Effectuez l'une des actions suivantes :
  - Pour convertir un script de FastExport tâche unique, développez le FastExport nœud, choisissez le script à convertir, puis choisissez Convertir le script dans le menu contextuel (clic droit).
  - Pour convertir plusieurs scripts, assurez-vous de sélectionner tous les scripts à convertir. Choisissez ensuite FastExport, ouvrez le menu contextuel (clic droit), puis choisissez Convertir le script.

AWS SCT convertit tous les scripts de FastExport travail Teradata que vous avez sélectionnés dans un format compatible avec Amazon Redshift RSQL. Trouvez vos scripts convertis dans le nœud Scripts du panneau de base de données cible.

5. Modifiez vos scripts Amazon Redshift RSQL convertis ou enregistrez-les. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de vos scripts de FastExport travail convertis](#).

## Gestion des scripts de FastExport travail Teradata avec AWS SCT

Vous pouvez ajouter plusieurs scripts de FastExport travail Teradata ou supprimer un script de FastExport travail de votre AWS SCT projet.

Pour ajouter un nouveau script de FastExport travail à votre AWS SCT projet

1. Développez le nœud Scripts dans le panneau de gauche.
2. Choisissez le FastExportnœud et ouvrez le menu contextuel (clic droit).
3. Choisissez Charger des scripts.
4. Entrez les informations requises pour ajouter un nouveau script de FastExport travail et configurer les variables de substitution. Pour plus d'informations, consultez [Ajouter des scripts de FastExport travail à votre AWS SCT projet](#) et [Configuration des variables de substitution dans les scripts de FastExport travail](#).

Pour supprimer un script de FastExport tâche de votre AWS SCT projet

1. Développez le FastExportnœud sous Scripts dans le panneau de gauche.
2. Choisissez le script à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer le script.

## Création d'un rapport d'évaluation pour la conversion d'un script de FastExport travail Teradata avec AWS SCT

Le rapport d'évaluation de la conversion des scripts de FastExport travail fournit des informations sur la conversion des FastExport commandes et des instructions SQL de vos FastExport scripts dans un format compatible avec Amazon Redshift RSQL. Le rapport d'évaluation inclut des actions pour FastExport les commandes et les instructions SQL qui ne AWS SCT peuvent pas être converties.

Pour créer un rapport d'évaluation de la conversion de scripts pour une tâche Teradata FastExport

1. Développez le FastExportnœud sous Scripts dans le panneau de gauche.

2. Choisissez le script à convertir, ouvrez le menu contextuel (clic droit), puis choisissez Créer un rapport.
3. Consultez l'onglet Résumé. L'onglet Résumé affiche les informations de synthèse issues du rapport d'évaluation du script de FastExport travail. Il inclut les résultats de conversion pour toutes les FastExport commandes et instructions SQL de vos scripts source.
4. Vous pouvez enregistrer une copie locale du rapport d'évaluation de la conversion du script de FastExport travail sous forme de fichier PDF ou de fichier CSV (valeurs séparées par des virgules).
  - a. Pour enregistrer le rapport d'évaluation de la conversion du script de FastExport tâche sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.

Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations pour la conversion des scripts.

- b. Pour enregistrer le rapport d'évaluation de la conversion du script de FastExport travail sous forme de fichier CSV, choisissez Enregistrer au format CSV en haut à droite.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.

5. Choisissez l'onglet Éléments d'action. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle vers Amazon Redshift RSQL. Lorsque vous sélectionnez une action dans la liste, mettez AWS SCT en surbrillance l'élément de votre script de FastExport tâche source auquel l'action s'applique.

## Modification et enregistrement de vos scripts de FastExport travail Teradata convertis avec AWS SCT

Vous pouvez modifier vos scripts convertis dans le panneau inférieur de votre AWS SCT projet. AWS SCT enregistre le script modifié dans le cadre de votre projet.

Pour enregistrer vos scripts convertis

1. Développez le nœud de scripts RSQL sous Scripts dans le panneau de base de données cible.
2. Choisissez votre script converti, ouvrez le menu contextuel (clic droit) et choisissez Enregistrer le script.
3. Entrez le chemin d'accès au dossier pour enregistrer le script converti et choisissez Enregistrer.

AWS SCT enregistre le script converti dans un fichier et ouvre ce fichier.

## Conversion de scripts de FastLoad travail en Amazon Redshift RSQL avec AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir les scripts de FastLoad travail Teradata en Amazon Redshift RSQL.

Un FastLoad script Teradata est un ensemble de commandes qui utilisent plusieurs sessions pour charger des données dans une table vide d'une base de données Teradata. Teradata FastLoad traite une série de FastLoad commandes Teradata et d'instructions SQL. Les FastLoad commandes Teradata permettent le contrôle de session et le traitement des données lors des transferts de données. Les instructions SQL créent, gèrent et suppriment des tables.

AWS SCT convertit FastLoad les commandes Teradata et les instructions SQL dans un format compatible avec Amazon Redshift RSQL. Après avoir migré la base de données Teradata vers Amazon Redshift, vous pouvez utiliser ces scripts convertis pour charger des données dans votre base de données Amazon Redshift.

### Rubriques

- [Ajouter des scripts de FastLoad travail à votre AWS SCT projet](#)
- [Configuration des variables de substitution dans les scripts de FastLoad travail Teradata avec AWS SCT](#)
- [Conversion de scripts de FastLoad travail Teradata avec AWS SCT](#)
- [Gestion des scripts de FastLoad travail Teradata avec AWS SCT](#)
- [Création d'un rapport d'évaluation pour la conversion d'un script de FastLoad travail Teradata avec AWS SCT](#)
- [Modification et enregistrement de vos scripts de FastLoad travail Teradata convertis avec AWS SCT](#)

## Ajouter des scripts de FastLoad travail à votre AWS SCT projet

Vous pouvez ajouter plusieurs scripts à un même AWS SCT projet.

## Pour ajouter un script de FastLoad tâche à votre AWS SCT projet

1. Créez un nouveau projet dans AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu, puis Teradata pour ajouter votre base de données source au projet. Pour de plus amples informations, veuillez consulter [Bases de données Teradata](#).
3. Choisissez Ajouter une cible dans le menu et ajoutez une base de données Amazon Redshift cible à votre AWS SCT projet.

Vous pouvez utiliser une plate-forme de base de données cible Amazon Redshift virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Créez une nouvelle règle de mappage qui inclut votre base de données Teradata source et votre cible Amazon Redshift. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
5. Dans le menu Affichage, choisissez Vue principale.
6. Dans le panneau de gauche, développez le nœud Scripts.
7. Choisissez FastLoad, ouvrez le menu contextuel (clic droit), puis choisissez Charger des scripts.
8. Entrez l'emplacement de vos scripts de FastLoad travail Teradata source et choisissez Sélectionner un dossier.

AWS SCT affiche la fenêtre Charger des scripts.

9. Effectuez l'une des actions suivantes :
  - Si vos scripts de FastLoad travail Teradata n'incluent pas les variables de substitution, choisissez Aucune variable de substitution, puis cliquez sur OK pour ajouter des scripts à votre AWS SCT projet.
  - Si vos scripts de FastLoad travail Teradata incluent des variables de substitution, configurez-les. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de FastLoad travail](#).

# Configuration des variables de substitution dans les scripts de FastLoad travail Teradata avec AWS SCT

Vos scripts de FastLoad travail Teradata peuvent inclure des variables de substitution. Par exemple, vous pouvez utiliser un seul script avec des variables de substitution pour charger des données dans différentes bases de données.

Avant d'exécuter un script de FastLoad travail avec des variables de substitution, assurez-vous d'attribuer les valeurs à toutes les variables. Pour ce faire, vous pouvez utiliser d'autres outils ou applications tels qu'un script Bash, UC4 (Automic), etc.

AWS SCT ne peut résoudre et convertir des variables de substitution qu'après avoir attribué leurs valeurs. Avant de commencer la conversion de vos scripts de FastLoad travail Teradata sources, assurez-vous d'attribuer des valeurs à toutes les variables de substitution. Vous pouvez l'utiliser AWS SCT pour configurer des variables de substitution dans vos scripts Teradata.

Pour configurer les variables de substitution dans votre script de FastLoad tâche

1. Lorsque vous ajoutez vos scripts de FastLoad travail Teradata source à votre AWS SCT projet, choisissez Substitution variables are used. Pour plus d'informations sur l'ajout de ces scripts, consultez [Ajouter des scripts de FastLoad travail à votre AWS SCT projet](#).
2. Pour Définir le format de variable, entrez une expression régulière correspondant à toutes les variables de substitution de votre script.

Par exemple, si les noms de vos variables de substitution commencent par `#{` et se terminent par `}`, utilisez l'expression `\#{\w+}` régulière. Pour faire correspondre les variables de substitution qui commencent par le signe dollar ou par le signe pourcentage, utilisez l'expression `[\$|%\w+]` régulière.

Expressions régulières AWS SCT conformes à la syntaxe des expressions régulières Java. Pour plus d'informations, consultez le [modèle de classe java.util.regex dans](#) la documentation Java.

3. Cliquez sur OK pour charger des scripts dans votre AWS SCT projet, puis sur OK pour fermer la fenêtre Charger des scripts.
4. Dans le panneau de gauche, développez le nœud Scripts. Choisissez FastLoad, puis choisissez votre dossier contenant les scripts. Ouvrez le menu contextuel (clic droit), puis choisissez Exporter les variables sous Variables de substitution.

Vous pouvez également exporter des variables de substitution pour un script. Complétez votre dossier avec des scripts, choisissez votre script, ouvrez le menu contextuel (clic droit) et choisissez Exporter les variables sous Variables de substitution.

5. Entrez le nom du fichier de valeurs séparées par des virgules (CSV) pour enregistrer les variables de substitution, puis choisissez Enregistrer.
6. Ouvrez ce fichier CSV et renseignez les valeurs des variables de substitution.

En fonction du système d'exploitation, AWS SCT utilise différents formats pour le fichier CSV. Les valeurs du fichier peuvent être placées entre guillemets ou non. Assurez-vous d'utiliser le même format pour les valeurs des variables de substitution que pour les autres valeurs du fichier. AWS SCT Impossible d'importer le fichier CSV contenant des valeurs dans différents formats.

7. Enregistrez le fichier CSV.
8. Dans le panneau de gauche, développez le nœud Scripts. Choisissez FastLoad, puis choisissez votre script. Ouvrez le menu contextuel (clic droit), puis choisissez Importer des variables sous Variables de substitution.
9. Choisissez votre fichier CSV, puis sélectionnez Ouvrir.
10. Choisissez Variables pour afficher toutes les variables de substitution découvertes et leurs valeurs.

## Conversion de scripts de FastLoad travail Teradata avec AWS SCT

Découvrez ci-dessous comment convertir une FastLoad tâche Teradata en Amazon Redshift RSQL à l'aide de. AWS SCT

Pour convertir un script de FastLoad tâche Teradata en Amazon Redshift RSQL

1. Ajoutez vos scripts de FastLoad travail à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts de FastLoad travail à votre AWS SCT projet](#).
2. Configurez les variables de substitution. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de FastLoad travail](#).
3. Dans le panneau de gauche, développez le nœud Scripts.
4. Effectuez l'une des actions suivantes :



- Pour convertir un script de FastLoad tâche unique, développez le FastLoadnœud, choisissez le script à convertir, puis choisissez Convertir le script dans le menu contextuel (clic droit).
- Pour convertir plusieurs scripts, assurez-vous de sélectionner tous les scripts à convertir. Choisissez FastLoad, ouvrez le menu contextuel (clic droit), puis choisissez Convertir le script. Ensuite, effectuez l'une des actions suivantes :
- Si vous stockez votre fichier de données source sur Amazon S3, choisissez le chemin de l'objet S3 pour l'emplacement du fichier de données source.

Entrez les valeurs du dossier du compartiment Amazon S3 et du compartiment Amazon S3 pour le fichier manifeste de votre fichier de données source.

- Si vous ne stockez pas votre fichier de données source sur Amazon S3, choisissez Host address pour l'emplacement du fichier de données source.

Entrez des valeurs pour l'URL ou l'adresse IP de l'hôte, le nom de connexion de l'utilisateur hôte et le compartiment Amazon S3 pour le fichier manifeste de votre fichier de données source.

5. Choisissez OK.

AWS SCT convertit tous les scripts de FastLoad travail Teradata que vous avez sélectionnés dans un format compatible avec Amazon Redshift RSQL. Trouvez vos scripts convertis dans le nœud Scripts du panneau de base de données cible.

6. Modifiez vos scripts Amazon Redshift RSQL convertis ou enregistrez-les. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de vos scripts de FastLoad travail convertis](#).

## Gestion des scripts de FastLoad travail Teradata avec AWS SCT

Vous pouvez ajouter plusieurs scripts de FastLoad travail Teradata ou supprimer un script de FastLoad travail de votre AWS SCT projet.

Pour ajouter un nouveau script de FastLoad travail à votre AWS SCT projet

1. Développez le nœud Scripts dans le panneau de gauche.
2. Choisissez le FastLoadnœud et ouvrez le menu contextuel (clic droit).
3. Choisissez Charger des scripts.

4. Entrez les informations requises pour ajouter un nouveau script de FastLoad travail et configurer les variables de substitution. Pour plus d'informations, consultez [Ajouter des scripts de FastLoad travail à votre AWS SCT projet](#) et [Configuration des variables de substitution dans les scripts de FastLoad travail](#).

Pour supprimer un script de FastLoad tâche de votre AWS SCT projet

1. Développez le FastLoadnœud sous Scripts dans le panneau de gauche.
2. Choisissez le script à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer le script.

## Création d'un rapport d'évaluation pour la conversion d'un script de FastLoad travail Teradata avec AWS SCT

Le rapport d'évaluation de la conversion des scripts de FastLoad travail fournit des informations sur la conversion FastLoad des commandes et des instructions SQL. La conversion s'effectue à partir de vos scripts source vers un format compatible avec Amazon Redshift RSQL. Le rapport d'évaluation inclut des actions pour FastLoad les commandes et les instructions SQL qui ne AWS SCT peuvent pas être converties.

Pour créer un rapport d'évaluation de la conversion de scripts pour une tâche Teradata FastLoad

1. Développez le FastLoadnœud sous Scripts dans le panneau de gauche.
2. Choisissez le script à convertir, ouvrez le menu contextuel (clic droit), puis choisissez Créer un rapport.
3. Consultez l'onglet Résumé.

L'onglet Résumé affiche les informations de synthèse issues du rapport d'évaluation du script de FastLoad travail. Il inclut les résultats de conversion pour toutes les FastLoad commandes et instructions SQL de vos scripts source.

4. (Facultatif) Enregistrez une copie locale du rapport d'évaluation de la conversion du script de FastLoad travail sous forme de fichier PDF ou de fichier CSV (valeurs séparées par des virgules) :
  - Pour enregistrer le rapport d'évaluation de la conversion du script de FastLoad tâche sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.

Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations relatives à la conversion des scripts.

- Pour enregistrer le rapport d'évaluation de la conversion du script de FastLoad travail sous forme de fichier CSV, choisissez Enregistrer au format CSV en haut à droite.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.

5. Choisissez l'onglet Éléments d'action. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle vers Amazon Redshift RSQL. Lorsque vous sélectionnez une action dans la liste, mettez AWS SCT en surbrillance l'élément de votre script de FastLoad tâche source auquel l'action s'applique.

## Modification et enregistrement de vos scripts de FastLoad travail Teradata convertis avec AWS SCT

Vous pouvez modifier vos scripts convertis dans le panneau inférieur de votre AWS SCT projet. AWS SCT enregistre le script modifié dans le cadre de votre projet.

Pour enregistrer vos scripts convertis

1. Développez le nœud de scripts RSQL sous Scripts dans le panneau de base de données cible.
2. Choisissez votre script converti, ouvrez le menu contextuel (clic droit) et choisissez Enregistrer le script.
3. Entrez le chemin d'accès au dossier pour enregistrer le script converti et choisissez Enregistrer.

AWS SCT enregistre le script converti dans un fichier et ouvre ce fichier.

## Conversion de MultiLoad scripts vers Amazon Redshift RSQL avec AWS Schema Conversion Tool

Vous pouvez l'utiliser AWS SCT pour convertir des scripts de MultiLoad travail Teradata en Amazon Redshift RSQL.

Un script de MultiLoad travail Teradata est un ensemble de commandes pour la maintenance par lots de votre base de données Teradata. Une tâche MultiLoad d'importation de Teradata effectue un

certain nombre d'opérations d'insertion, de mise à jour et de suppression sur un maximum de cinq tables et vues différentes. Les tâches de MultiLoad suppression de Teradata peuvent supprimer un grand nombre de lignes d'une même table.

AWS SCT convertit MultiLoad les commandes Teradata et les instructions SQL dans un format compatible avec Amazon Redshift RSQL. Après avoir migré la base de données Teradata vers Amazon Redshift, utilisez ces scripts convertis pour gérer les données de votre base de données Amazon Redshift.

## Rubriques

- [Ajouter des scripts de MultiLoad travail à votre AWS SCT projet](#)
- [Configuration des variables de substitution dans les scripts de MultiLoad travail Teradata avec AWS SCT](#)
- [Conversion de scripts de MultiLoad travail Teradata avec AWS SCT](#)
- [Gestion des scripts de MultiLoad travail Teradata avec AWS SCT](#)
- [Création d'un rapport d'évaluation pour la conversion d'un script de MultiLoad travail Teradata avec AWS SCT](#)
- [Modification et enregistrement de vos scripts de MultiLoad travail Teradata convertis avec AWS SCT](#)

## Ajouter des scripts de MultiLoad travail à votre AWS SCT projet

Vous pouvez ajouter plusieurs scripts à un même AWS SCT projet.

Pour ajouter un script de MultiLoad tâche à votre AWS SCT projet

1. Créez un nouveau projet AWS SCT ou ouvrez un projet existant. Pour de plus amples informations, veuillez consulter [the section called “Démarrage et gestion de projets”](#).
2. Choisissez Ajouter une source dans le menu, puis Teradata pour ajouter votre base de données source au projet. Pour de plus amples informations, veuillez consulter [Bases de données Teradata](#).
3. Choisissez Ajouter une cible dans le menu pour ajouter une base de données Amazon Redshift cible à votre AWS SCT projet.

Vous pouvez utiliser une plate-forme de base de données cible Amazon Redshift virtuelle. Pour de plus amples informations, veuillez consulter [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

4. Créez une nouvelle règle de mappage qui inclut votre base de données Teradata source et votre cible Amazon Redshift. Pour de plus amples informations, veuillez consulter [Cartographie de nouveaux types de données dans AWS Schema Conversion Tool](#).
5. Dans le menu Affichage, choisissez Vue principale.
6. Dans le panneau de gauche, développez le nœud Scripts.
7. Choisissez MultiLoad, ouvrez le menu contextuel (clic droit), puis choisissez Charger des scripts.
8. Entrez l'emplacement de vos scripts de MultiLoad travail Teradata source et choisissez Sélectionner un dossier.

AWS SCT affiche la fenêtre Charger des scripts.

9. Effectuez l'une des actions suivantes :
  - Si vos scripts de MultiLoad travail Teradata n'incluent pas les variables de substitution, choisissez Aucune variable de substitution, puis cliquez sur OK pour ajouter des scripts à votre AWS SCT projet.
  - Si vos scripts de MultiLoad travail Teradata incluent des variables de substitution, configurez-les. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de MultiLoad travail](#).

## Configuration des variables de substitution dans les scripts de MultiLoad travail Teradata avec AWS SCT

Vos scripts de MultiLoad travail Teradata peuvent inclure des variables de substitution. Par exemple, vous pouvez utiliser un seul script avec des variables de substitution pour charger des données dans différentes bases de données.

Avant d'exécuter un script de MultiLoad travail avec des variables de substitution, assurez-vous d'attribuer les valeurs à toutes les variables. Pour ce faire, vous pouvez utiliser d'autres outils ou applications tels qu'un script Bash, UC4 (Automic), etc.

AWS SCT ne peut résoudre et convertir des variables de substitution qu'après avoir attribué leurs valeurs. Avant de commencer la conversion de vos scripts de MultiLoad travail Teradata sources,

assurez-vous d'avoir attribué des valeurs à toutes les variables de substitution. Vous pouvez l'utiliser AWS SCT pour configurer des variables de substitution dans vos scripts Teradata.

Pour configurer les variables de substitution dans votre script de MultiLoad tâche

1. Lorsque vous ajoutez vos scripts de MultiLoad travail Teradata source à votre AWS SCT projet, choisissez Substitution variables are used. Pour plus d'informations sur l'ajout de ces scripts, consultez [Ajouter des scripts de MultiLoad travail à votre AWS SCT projet](#).
2. Pour Définir le format de variable, entrez une expression régulière correspondant à toutes les variables de substitution de votre script.

Par exemple, si les noms de vos variables de substitution commencent par `#{` et se terminent par `}`, utilisez l'expression `\#{\w+}` régulière. Pour faire correspondre les variables de substitution qui commencent par le signe dollar ou par le signe pourcentage, utilisez l'expression `[\$|%\w+]` régulière.

Expressions régulières AWS SCT conformes à la syntaxe des expressions régulières Java. Pour plus d'informations, consultez le [modèle de classe java.util.regex](#) dans la documentation Java.

3. Cliquez sur OK pour charger des scripts dans votre AWS SCT projet, puis sur OK pour fermer la fenêtre Charger des scripts.
4. Choisissez Variables pour afficher toutes les variables de substitution découvertes et leurs valeurs.
5. Dans Valeur, entrez la valeur de la variable de substitution.

## Conversion de scripts de MultiLoad travail Teradata avec AWS SCT

Découvrez ci-dessous comment convertir une MultiLoad tâche Teradata en Amazon Redshift RSQL à l'aide de AWS SCT

Pour convertir un script de MultiLoad tâche Teradata en Amazon Redshift RSQL

1. Ajoutez vos scripts de MultiLoad travail à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Ajouter des scripts de MultiLoad travail à votre AWS SCT projet](#).
2. Configurez les variables de substitution et entrez leurs valeurs. Pour de plus amples informations, veuillez consulter [Configuration des variables de substitution dans les scripts de MultiLoad travail](#).
3. Dans le panneau de gauche, développez le nœud Scripts.

#### 4. Effectuez l'une des actions suivantes :

- Pour convertir un script de MultiLoad tâche unique, développez le MultiLoadnœud, choisissez le script à convertir, puis choisissez Convertir le script dans le menu contextuel (clic droit).
- Pour convertir plusieurs scripts, assurez-vous de sélectionner tous les scripts à convertir. Choisissez MultiLoad, ouvrez le menu contextuel (clic droit), puis choisissez Convertir le script.

#### 5. Effectuez l'une des actions suivantes :

- Si vous stockez votre fichier de données source sur Amazon S3, choisissez le chemin de l'objet S3 pour l'emplacement du fichier de données source.

Entrez le dossier du compartiment Amazon S3 et le compartiment Amazon S3 comme fichier manifeste pour votre fichier de données source.

- Si vous ne stockez pas votre fichier de données source sur Amazon S3, choisissez Host address pour l'emplacement du fichier de données source.

Entrez l'URL ou l'adresse IP de l'hôte, le nom de connexion de l'utilisateur hôte et le compartiment Amazon S3 pour le fichier manifeste de votre fichier de données source.

#### 6. Choisissez OK.

AWS SCT convertit tous les scripts de MultiLoad travail Teradata que vous avez sélectionnés dans un format compatible avec Amazon Redshift RSQL. Trouvez vos scripts convertis dans le nœud Scripts du panneau de base de données cible.

#### 7. Modifiez vos scripts Amazon Redshift RSQL convertis ou enregistrez-les. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de vos scripts de MultiLoad travail convertis](#).

## Gestion des scripts de MultiLoad travail Teradata avec AWS SCT

Vous pouvez ajouter plusieurs scripts de MultiLoad travail Teradata ou supprimer un script de MultiLoad travail de votre AWS SCT projet.

Pour ajouter un nouveau script de MultiLoad travail à votre AWS SCT projet

1. Développez le nœud Scripts dans le panneau de gauche.
2. Choisissez le MultiLoadnœud et ouvrez le menu contextuel (clic droit).
3. Choisissez Charger des scripts.

4. Entrez les informations requises pour ajouter un nouveau script de MultiLoad travail et configurer les variables de substitution. Pour plus d'informations, consultez [Ajouter des scripts de MultiLoad travail à votre AWS SCT projet](#) et [Configuration des variables de substitution dans les scripts de MultiLoad travail](#).

Pour supprimer un script de MultiLoad tâche de votre AWS SCT projet

1. Développez le MultiLoadnœud sous Scripts dans le panneau de gauche.
2. Choisissez le script à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer le script.

## Création d'un rapport d'évaluation pour la conversion d'un script de MultiLoad travail Teradata avec AWS SCT

Le rapport d'évaluation de la conversion des scripts de MultiLoad travail fournit des informations sur la conversion MultiLoad des commandes et des instructions SQL. La conversion s'effectue à partir de vos scripts source en commandes RSQL Amazon Redshift et en instructions SQL pour Amazon Redshift. Le rapport d'évaluation inclut des actions pour MultiLoad les commandes et les instructions SQL qui ne AWS SCT peuvent pas être converties.

Pour créer un rapport d'évaluation de la conversion de scripts pour une tâche Teradata MultiLoad

1. Développez le MultiLoadnœud sous Scripts dans le panneau de gauche.
2. Choisissez les scripts pour lesquels créer le rapport d'évaluation, ouvrez le menu contextuel (clic droit), puis choisissez Créer un rapport.
3. Consultez l'onglet Résumé. L'onglet Résumé affiche les informations de synthèse issues du rapport d'évaluation du script de MultiLoad travail. Il inclut les résultats de conversion pour toutes les MultiLoad commandes et instructions SQL de vos scripts source.
4. (Facultatif) Enregistrez une copie locale du rapport d'évaluation de la conversion du script de MultiLoad travail sous forme de fichier PDF ou de fichiers CSV (valeurs séparées par des virgules) :
  - Pour enregistrer le rapport d'évaluation de la conversion du script de MultiLoad tâche sous forme de fichier PDF, choisissez Enregistrer au format PDF en haut à droite.



Le fichier PDF contient le résumé, les actions à entreprendre et les recommandations pour la conversion des scripts.

- Pour enregistrer le rapport d'évaluation de la conversion du script de MultiLoad travail sous forme de fichiers CSV, choisissez Enregistrer au format CSV en haut à droite.

AWS SCT crée deux fichiers CSV. Ces fichiers contiennent le résumé, les mesures à prendre, les actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir les scripts.

5. Choisissez l'onglet Éléments d'action. Cet onglet contient la liste des éléments qui nécessitent une conversion manuelle vers Amazon Redshift RSQL. Lorsque vous sélectionnez une action dans la liste, mettez AWS SCT en surbrillance l'élément de votre script de MultiLoad tâche source auquel l'action s'applique.

## Modification et enregistrement de vos scripts de MultiLoad travail Teradata convertis avec AWS SCT

Vous pouvez modifier vos scripts convertis dans le panneau inférieur de votre AWS SCT projet. AWS SCT enregistre le script modifié dans le cadre de votre projet.

Pour enregistrer vos scripts convertis

1. Développez le nœud de scripts RSQL sous Scripts dans le panneau de base de données cible.
2. Choisissez votre script converti, ouvrez le menu contextuel (clic droit) et choisissez Enregistrer le script.
3. Entrez le chemin d'accès au dossier pour enregistrer le script converti et choisissez Enregistrer.

AWS SCT enregistre le script converti dans un fichier et ouvre ce fichier.

# Migrer des infrastructures de mégadonnées avec AWS Schema Conversion Tool

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour migrer des frameworks de mégadonnées vers le AWS Cloud.

Actuellement, AWS SCT prend en charge la migration des clusters Hadoop vers Amazon EMR et Amazon S3. Ce processus de migration inclut les services Hive et HDFS.

Vous pouvez également l'utiliser AWS SCT pour automatiser la conversion de vos flux de travail d'orchestration Apache Oozie en AWS Step Functions

## Rubriques

- [Migration des charges de travail Hadoop vers Amazon EMR avec AWS Schema Conversion Tool](#)
- [Convertir le flux de travail Oozie en with AWS Step FunctionsAWS Schema Conversion Tool](#)

## Migration des charges de travail Hadoop vers Amazon EMR avec AWS Schema Conversion Tool

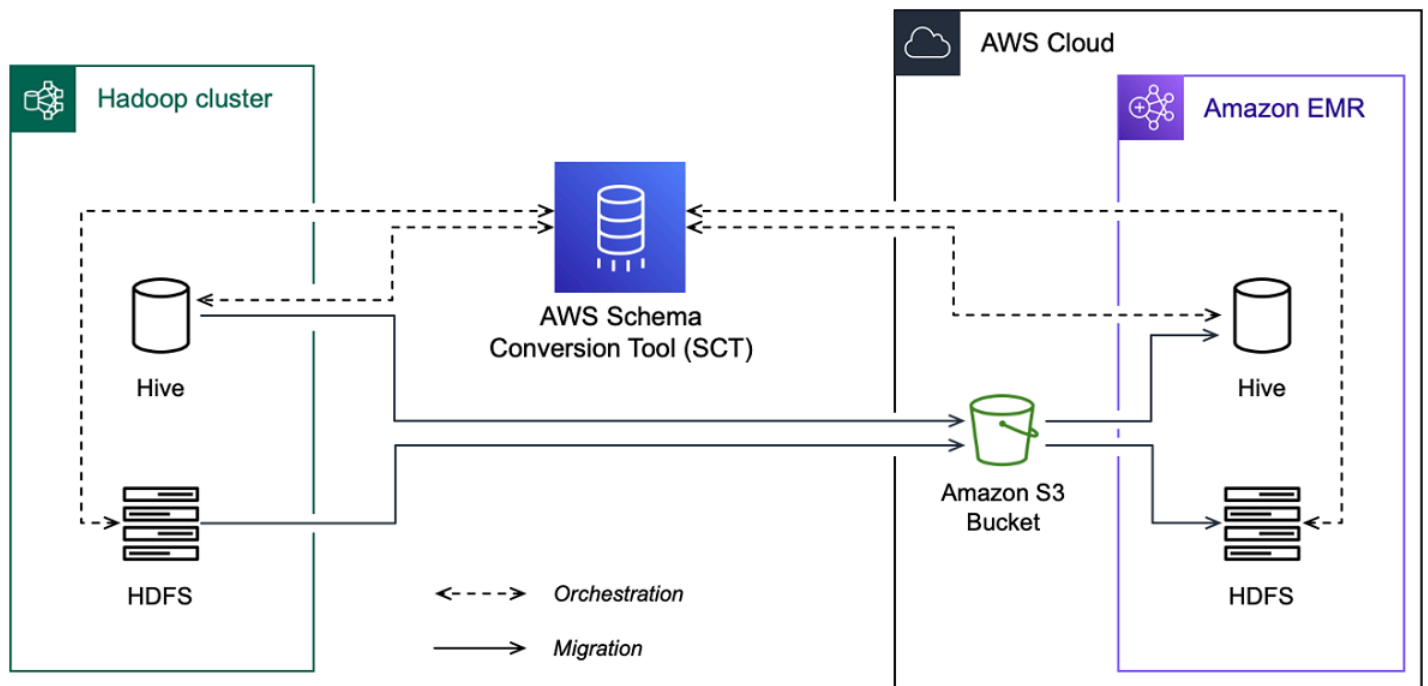
Pour migrer des clusters Apache Hadoop, assurez-vous d'utiliser la AWS SCT version 1.0.670 ou supérieure. Familiarisez-vous également avec l'interface de ligne de commande (CLI) de AWS SCT. Pour de plus amples informations, veuillez consulter [Référence CLI pour AWS Schema Conversion Tool](#).

## Rubriques

- [Présentation de la migration](#)
- [Étape 1 : Connectez-vous à vos clusters Hadoop](#)
- [Étape 2 : configurer les règles de mappage](#)
- [Étape 3 : Création d'un rapport d'évaluation](#)
- [Étape 4 : migrer votre cluster Apache Hadoop vers Amazon EMR avec AWS SCT](#)
- [Exécution de votre script CLI](#)
- [Gestion de votre projet de migration de mégadonnées](#)

## Présentation de la migration

L'image suivante montre le schéma d'architecture de la migration d'Apache Hadoop vers Amazon EMR.



AWS SCT migre les données et les métadonnées de votre cluster Hadoop source vers un compartiment Amazon S3. Ensuite, AWS SCT utilise vos métadonnées Hive source pour créer des objets de base de données dans le service Amazon EMR Hive cible. Vous pouvez éventuellement configurer Hive pour qu'il utilise le AWS Glue Data Catalog comme métastore. Dans ce cas, AWS SCT migre vos métadonnées Hive source vers le AWS Glue Data Catalog.

Vous pouvez ensuite les utiliser AWS SCT pour migrer les données d'un compartiment Amazon S3 vers votre service Amazon EMR HDFS cible. Vous pouvez également laisser les données dans votre compartiment Amazon S3 et les utiliser comme référentiel de données pour vos charges de travail Hadoop.

Pour démarrer la migration vers Hadoop, vous devez créer et exécuter votre script AWS SCT CLI. Ce script inclut l'ensemble complet des commandes permettant d'exécuter la migration. Vous pouvez télécharger et modifier un modèle de script de migration Hadoop. Pour de plus amples informations, veuillez consulter [Obtenir des scénarios CLI](#).

Assurez-vous que votre script inclut les étapes suivantes afin de pouvoir exécuter votre migration d'Apache Hadoop vers Amazon S3 et Amazon EMR.

## Étape 1 : Connectez-vous à vos clusters Hadoop

Pour démarrer la migration de votre cluster Apache Hadoop, créez un nouveau AWS SCT projet. Connectez-vous ensuite à vos clusters source et cible. Assurez-vous de créer et de provisionner vos AWS ressources cibles avant de commencer la migration.

Au cours de cette étape, vous devez utiliser les commandes AWS SCT CLI suivantes.

- `CreateProject`— pour créer un nouveau AWS SCT projet.
- `AddSourceCluster`— pour vous connecter au cluster Hadoop source de votre AWS SCT projet.
- `AddSourceClusterHive`— pour vous connecter au service Hive source de votre projet.
- `AddSourceClusterHDFS`— pour vous connecter au service HDFS source de votre projet.
- `AddTargetCluster`— pour vous connecter au cluster Amazon EMR cible de votre projet.
- `AddTargetClusterS3`— pour ajouter le compartiment Amazon S3 à votre projet.
- `AddTargetClusterHive`— pour vous connecter au service Hive cible de votre projet
- `AddTargetClusterHDFS`— pour vous connecter au service HDFS cible de votre projet

Pour des exemples d'utilisation de ces commandes AWS SCT CLI, consultez [Connexion à Apache Hadoop](#).

Lorsque vous exécutez la commande qui se connecte à un cluster source ou cible, AWS SCT essaie d'établir la connexion à ce cluster. Si la tentative de connexion échoue, AWS SCT arrête d'exécuter les commandes de votre script CLI et affiche un message d'erreur.

## Étape 2 : configurer les règles de mappage

Après vous être connecté à vos clusters source et cible, configurez les règles de mappage. Une règle de mappage définit la cible de migration pour un cluster source. Assurez-vous de définir des règles de mappage pour tous les clusters sources que vous avez ajoutés dans votre AWS SCT projet. Pour plus d'informations sur les règles de mappage, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Dans cette étape, vous devez utiliser la `AddServerMapping` commande. Cette commande utilise deux paramètres qui définissent les clusters source et cible. Vous pouvez utiliser la `AddServerMapping` commande avec le chemin explicite vers les objets de votre base de données ou avec le nom d'un objet. Pour la première option, vous devez inclure le type de l'objet et son nom. Pour la deuxième option, vous n'incluez que les noms des objets.

- `sourceTreePath`— le chemin explicite vers les objets de votre base de données source.  
`targetTreePath`— le chemin explicite vers les objets de votre base de données cible.
- `sourceNamePath`: le chemin qui inclut uniquement les noms de vos objets source.  
`targetNamePath`: le chemin qui inclut uniquement les noms de vos objets cibles.

L'exemple de code suivant crée une règle de mappage utilisant des chemins explicites pour la base de données `testdb` Hive source et le cluster EMR cible.

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Vous pouvez utiliser cet exemple et les exemples suivants sous Windows. Pour exécuter les commandes CLI sous Linux, assurez-vous d'avoir mis à jour les chemins de fichiers en fonction de votre système d'exploitation.

L'exemple de code suivant crée une règle de mappage en utilisant les chemins qui incluent uniquement les noms des objets.

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Vous pouvez choisir Amazon EMR ou Amazon S3 comme cible pour votre objet source. Pour chaque objet source, vous ne pouvez choisir qu'une seule cible dans un seul AWS SCT projet. Pour modifier la cible de migration d'un objet source, supprimez la règle de mappage existante, puis créez-en une nouvelle. Pour supprimer une règle de mappage, utilisez la `DeleteServerMapping` commande. Cette commande utilise l'un des deux paramètres suivants.

- `sourceTreePath`— le chemin explicite vers les objets de votre base de données source.
- `sourceNamePath`: le chemin qui inclut uniquement les noms de vos objets source.

Pour plus d'informations sur les `DeleteServerMapping` commandes `AddServerMapping` et, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

## Étape 3 : Création d'un rapport d'évaluation

Avant de commencer la migration, nous vous recommandons de créer un rapport d'évaluation. Ce rapport résume toutes les tâches de migration et détaille les actions qui apparaîtront au cours de la migration. Pour vous assurer que votre migration n'échoue pas, consultez ce rapport et abordez les mesures à prendre avant la migration. Pour de plus amples informations, veuillez consulter [Rapport d'évaluation](#).

Dans cette étape, vous devez utiliser la `CreateMigrationReport` commande. Cette commande utilise deux paramètres. Le `treePath` paramètre est obligatoire, tandis que le `forceMigrate` paramètre est facultatif.

- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `forceMigrate`— lorsque ce paramètre est défini sur `true`, AWS SCT poursuit la migration même si votre projet inclut un dossier HDFS et une table Hive faisant référence au même objet. La valeur par défaut est `false`.

Vous pouvez ensuite enregistrer une copie du rapport d'évaluation au format PDF ou au format CSV (valeurs séparées par des virgules). Pour ce faire, utilisez la `SaveReportCSV` commande `SaveReportPDF` or.

La `SaveReportPDF` commande enregistre une copie de votre rapport d'évaluation sous forme de fichier PDF. Cette commande utilise quatre paramètres. Le `file` paramètre est obligatoire, les autres paramètres sont facultatifs.

- `file`— le chemin d'accès au fichier PDF et son nom.
- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `namePath`: le chemin qui inclut uniquement les noms des objets cibles pour lesquels vous enregistrez une copie du rapport d'évaluation.

La `SaveReportCSV` commande enregistre votre rapport d'évaluation dans trois fichiers CSV. Cette commande utilise quatre paramètres. Le `directory` paramètre est obligatoire, les autres paramètres sont facultatifs.

- `directory`— le chemin d'accès au dossier dans lequel AWS SCT sont enregistrés les fichiers CSV.
- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `namePath`: le chemin qui inclut uniquement les noms des objets cibles pour lesquels vous enregistrez une copie du rapport d'évaluation.

L'exemple de code suivant enregistre une copie du rapport d'évaluation dans le `c:\sct\ar.pdf` fichier.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

L'exemple de code suivant enregistre une copie du rapport d'évaluation sous forme de fichiers CSV dans le `c:\sct` dossier.

```
SaveReportCSV
-file:'c:\sct'
/
```

Pour plus d'informations sur les `SaveReportCSV` commandes `SaveReportPDF` et, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

## Étape 4 : migrer votre cluster Apache Hadoop vers Amazon EMR avec AWS SCT

Après avoir configuré votre AWS SCT projet, lancez la migration de votre cluster Apache Hadoop local vers le AWS Cloud

Au cours de cette étape, vous devez utiliser les `ResumeMigration` commandes `MigrateMigrationStatus`, et.

La `Migrate` commande fait migrer vos objets source vers le cluster cible. Cette commande utilise quatre paramètres. Assurez-vous de spécifier le `treePath` paramètre `filter` ou. Les autres paramètres sont facultatifs.

- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `forceLoad`— lorsqu'il est défini sur `true`, charge AWS SCT automatiquement les arborescences de métadonnées de la base de données lors de la migration. La valeur par défaut est `false`.
- `forceMigrate`— lorsque ce paramètre est défini sur `true`, AWS SCT poursuit la migration même si votre projet inclut un dossier HDFS et une table Hive faisant référence au même objet. La valeur par défaut est `false`.

La `MigrationStatus` commande renvoie les informations relatives à la progression de la migration. Pour exécuter cette commande, entrez le nom de votre projet de migration pour le `name` paramètre. Vous avez indiqué ce nom dans la `CreateProject` commande.

La `ResumeMigration` commande reprend la migration interrompue que vous avez lancée à l'aide de la `Migrate` commande. La `ResumeMigration` commande n'utilise pas de paramètres. Pour reprendre la migration, vous devez vous connecter à vos clusters source et cible. Pour de plus amples informations, veuillez consulter [Gestion de votre projet de migration](#).

L'exemple de code suivant migre les données de votre service HDFS source vers Amazon EMR.

```
Migrate
-treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
-forceMigrate: 'true'
/
```

## Exécution de votre script CLI

Une fois que vous avez terminé de modifier votre script AWS SCT CLI, enregistrez-le sous forme de fichier avec l'.`sct` extension. Vous pouvez maintenant exécuter votre script depuis le app dossier correspondant à votre chemin AWS SCT d'installation. Pour cela, utilisez la commande suivante.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Dans l'exemple précédent, remplacez `script_path` par le chemin d'accès à votre fichier par le script CLI. Pour plus d'informations sur l'exécution de scripts CLI dans AWS SCT, consultez [Mode script](#).



## Gestion de votre projet de migration de mégadonnées

Une fois la migration terminée, vous pouvez enregistrer et modifier votre AWS SCT projet pour une utilisation future.

Pour enregistrer votre AWS SCT projet, utilisez la `SaveProject` commande. Cette commande n'utilise pas de paramètres.

L'exemple de code suivant enregistre votre AWS SCT projet.

```
SaveProject  
/
```

Pour ouvrir votre AWS SCT projet, utilisez la `OpenProject` commande. Cette commande utilise un paramètre obligatoire. Pour le `file` paramètre, entrez le chemin d'accès à votre fichier de AWS SCT projet et son nom. Vous avez indiqué le nom du projet dans la `CreateProject` commande. Assurez-vous d'ajouter l'.sctsextension au nom de votre fichier de projet pour exécuter la `OpenProject` commande.

L'exemple de code suivant ouvre le `hadoop_emr` projet depuis le `c:\sct` dossier.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Après avoir ouvert votre AWS SCT projet, vous n'avez pas besoin d'ajouter les clusters source et cible, car vous les avez déjà ajoutés à votre projet. Pour commencer à travailler avec vos clusters source et cible, vous devez vous y connecter. Pour ce faire, vous devez utiliser les `ConnectTargetCluster` commandes `ConnectSourceCluster` et. Ces commandes utilisent les mêmes paramètres que les `AddTargetCluster` commandes `AddSourceCluster` et. Vous pouvez modifier votre script CLI et remplacer le nom de ces commandes en laissant la liste des paramètres inchangée.

L'exemple de code suivant permet de se connecter au cluster Hadoop source.

```
ConnectSourceCluster  
-name: 'HADOOP_SOURCE'  
-vendor: 'HADOOP'  
-host: 'hadoop_address'
```

```
-port: '22'  
-user: 'hadoop_user'  
-password: 'hadoop_password'  
-useSSL: 'true'  
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: 'hadoop_passphrase'
```

```
/
```

L'exemple de code suivant permet de se connecter au cluster Amazon EMR cible.

```
ConnectTargetCluster  
-name: 'HADOOP_TARGET'  
-vendor: 'AMAZON_EMR'  
-host: 'ec2-44-44-55-66.eu-west-1.EXAMPLE.amazonaws.com'  
-port: '22'  
-user: 'emr_user'  
-password: 'emr_password'  
-useSSL: 'true'  
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: '1234567890abcdef0!'  
-s3Name: 'S3_TARGET'  
-accessKey: 'AKIAIOSFODNN7EXAMPLE'  
-secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'  
-region: 'eu-west-1'  
-s3Path: 'doc-example-bucket/example-folder'
```

```
/
```

Dans l'exemple précédent, remplacez *hadoop\_address* par l'adresse IP de votre cluster Hadoop. Si nécessaire, configurez la valeur de la variable de port. Ensuite, remplacez *hadoop\_user* et *hadoop\_password* par le nom de votre utilisateur Hadoop et le mot de passe de cet utilisateur. Pour *path\name*, entrez le nom et le chemin du fichier PEM de votre cluster Hadoop source. Pour plus d'informations sur l'ajout de vos clusters source et cible, consultez [Connexion aux bases de données Apache Hadoop à l'aide du AWS Schema Conversion Tool](#).

Après vous être connecté à vos clusters Hadoop source et cible, vous devez vous connecter à vos services Hive et HDFS, ainsi qu'à votre compartiment Amazon S3. Pour ce faire, vous devez utiliser les `ConnectTargetClusterS3` commandes `ConnectSourceClusterHive` `ConnectSourceClusterHdfs` `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs`, et. Ces commandes utilisent les mêmes paramètres que les commandes que vous avez utilisées pour ajouter les services Hive et HDFS, ainsi que le bucket Amazon S3 à votre projet. Modifiez le script CLI pour remplacer le `Add` préfixe par `Connect` le nom des commandes.

# Convertir le flux de travail Oozie en with AWS Step FunctionsAWS Schema Conversion Tool

Pour convertir les flux de travail Apache Oozie, assurez-vous d'utiliser la AWS SCT version 1.0.671 ou supérieure. Familiarisez-vous également avec l'interface de ligne de commande (CLI) de AWS SCT. Pour de plus amples informations, veuillez consulter [Référence CLI pour AWS Schema Conversion Tool](#).

## Rubriques

- [Vue d'ensemble des conversions](#)
- [Étape 1 : Connectez-vous à vos services source et cible](#)
- [Étape 2 : configurer les règles de mappage](#)
- [Étape 3 : Configuration des paramètres](#)
- [Étape 4 : Créer un rapport d'évaluation](#)
- [Étape 5 : Convertissez vos flux de travail Apache Oozie en AWS Step FunctionsAWS SCT](#)
- [Exécution de votre script CLI](#)
- [Nœuds Apache Oozie AWS SCT pouvant être convertis en AWS Step Functions](#)

## Vue d'ensemble des conversions

Votre code source Apache Oozie inclut des nœuds d'action, des nœuds de flux de contrôle et des propriétés de tâche. Les nœuds d'action définissent les tâches que vous exécutez dans votre flux de travail Apache Oozie. Lorsque vous utilisez Apache Oozie pour orchestrer votre cluster Apache Hadoop, un nœud d'action inclut une tâche Hadoop. Les nœuds de flux de contrôle fournissent un mécanisme permettant de contrôler le chemin du flux de travail. Les nœuds du flux de contrôle incluent des nœuds tels que `startend`, `decision`, `fork`, et `join`.

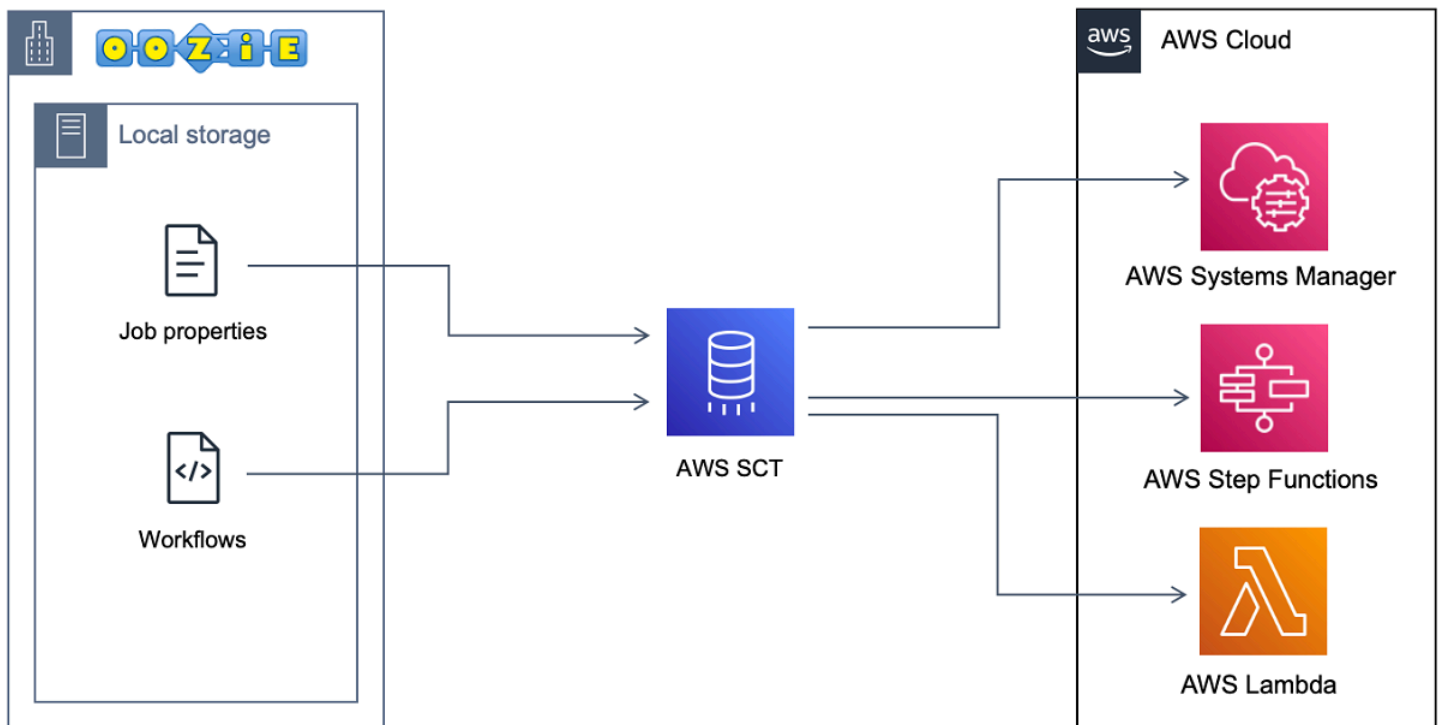
AWS SCT convertit vos nœuds d'action source et vos nœuds de flux de contrôle en AWS Step Functions. Dans AWS Step Functions, vous définissez vos flux de travail dans l'Amazon States Language (ASL). AWS SCT utilise l'ASL pour définir votre machine à états, qui est un ensemble d'états, capable de fonctionner, de déterminer les états vers lesquels passer ensuite, de s'arrêter en cas d'erreur, etc. Ensuite, AWS SCT télécharge les fichiers JSON avec les définitions des machines à états. AWS SCT Vous pouvez ensuite utiliser votre rôle AWS Identity and Access Management (IAM) pour configurer vos machines d'état dans AWS Step Functions. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Step Functions ?](#) dans le Guide AWS Step Functions du développeur.

AWS SCT Crée également un pack d'extension avec des AWS Lambda fonctions qui émulent les fonctions source qui AWS Step Functions ne sont pas prises en charge. Pour de plus amples informations, veuillez consulter [Utilisation de packs d'extension avec AWS Schema Conversion Tool](#).

AWS SCT migre les propriétés de votre tâche source vers. AWS Systems Manager Pour stocker les noms et les valeurs des paramètres, AWS SCT utilise Parameter Store, une fonctionnalité de AWS Systems Manager. Pour plus d'informations, voir [Qu'est-ce que c'est AWS Systems Manager ?](#) dans le guide de AWS Systems Manager l'utilisateur.

Vous pouvez l'utiliser AWS SCT pour mettre à jour automatiquement les valeurs et les noms de vos paramètres. En raison des différences d'architecture entre Apache Oozie et Apache AWS Step Functions, vous devrez peut-être configurer vos paramètres. AWS SCT peut trouver le nom ou la valeur d'un paramètre spécifié dans vos fichiers source et les remplacer par de nouvelles valeurs. Pour de plus amples informations, veuillez consulter [Étape 3 : Configuration des paramètres](#).

L'image suivante montre le schéma d'architecture de la conversion d'Apache Oozie en. AWS Step Functions



Pour démarrer la conversion, créez et exécutez votre script AWS SCT CLI. Ce script inclut l'ensemble complet des commandes permettant d'exécuter la conversion. Vous pouvez télécharger et modifier un modèle du script de conversion Apache Oozie. Pour de plus amples informations, veuillez consulter [Obtenir des scénarios CLI](#).

Assurez-vous que votre script inclut les étapes suivantes.

## Étape 1 : Connectez-vous à vos services source et cible

Pour démarrer la conversion de votre cluster Apache Oozie, créez un nouveau AWS SCT projet. Connectez-vous ensuite à vos services source et cible. Assurez-vous de créer et de provisionner vos AWS ressources cibles avant de commencer la migration. Pour de plus amples informations, veuillez consulter [Conditions préalables à l'utilisation d'Apache Oozie en tant que source](#).

Au cours de cette étape, vous devez utiliser les commandes AWS SCT CLI suivantes.

- `CreateProject`— pour créer un nouveau AWS SCT projet.
- `AddSource`— pour ajouter vos fichiers source Apache Oozie dans votre AWS SCT projet.
- `ConnectSource`— pour se connecter à Apache Oozie en tant que source.
- `AddTarget`— à ajouter AWS Step Functions en tant que cible de migration dans votre projet.
- `ConnectTarget`— pour se connecter à AWS Step Functions.

Pour des exemples d'utilisation de ces commandes AWS SCT CLI, consultez [Connexion à Apache Oozie](#).

Lorsque vous exécutez les `ConnectTarget` commandes `ConnectSource` or, AWS SCT essaie d'établir la connexion à vos services. Si la tentative de connexion échoue, AWS SCT arrête d'exécuter les commandes de votre script CLI et affiche un message d'erreur.

## Étape 2 : configurer les règles de mappage

Après vous être connecté à vos services source et cible, configurez les règles de mappage. Une règle de mappage définit la cible de migration pour vos flux de travail et paramètres Apache Oozie source. Pour plus d'informations sur les règles de mappage, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Pour définir les objets source et cible à convertir, utilisez la `AddServerMapping` commande. Cette commande utilise deux paramètres : `sourceTreePath` et `targetTreePath`. Les valeurs de ces paramètres incluent un chemin explicite vers vos objets source et cible. Pour qu'Apache Oozie soit AWS Step Functions converti, ces paramètres doivent commencer par `ETL`

L'exemple de code suivant crée une règle de mappage pour `OOZIE` et `AWS_STEP_FUNCTIONS` les objets. Vous avez ajouté ces objets à votre AWS SCT projet à l'aide `AddSource` des `AddTarget` commandes et à l'étape précédente.

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Pour plus d'informations sur la `AddServerMapping` commande, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

## Étape 3 : Configuration des paramètres

Si vos flux de travail Apache Oozie source utilisent des paramètres, vous devrez peut-être modifier leurs valeurs après la conversion en AWS Step Functions. Il se peut également que vous deviez ajouter de nouveaux paramètres à utiliser avec votre AWS Step Functions.

Au cours de cette étape, vous devez utiliser les `AddTargetParameter` commandes `AddParameterMapping` et.

Pour remplacer les valeurs des paramètres dans vos fichiers source, utilisez la `AddParameterMapping` commande. AWS SCT analyse vos fichiers source, trouve les paramètres par nom ou par valeur et modifie leurs valeurs. Vous pouvez exécuter une seule commande pour analyser tous vos fichiers sources. Vous définissez l'étendue des fichiers à analyser à l'aide de l'un des trois premiers paramètres de la liste suivante. Cette commande utilise jusqu'à six paramètres.

- `filterName`— le nom du filtre pour vos objets source. Vous pouvez créer un filtre à l'aide de la `CreateFilter` commande.
- `treePath`— le chemin explicite vers vos objets source.
- `namePath`— le chemin explicite vers un objet source spécifique.
- `sourceParameterName`— le nom de votre paramètre source.
- `sourceValue`— la valeur de votre paramètre source.
- `targetValue`— la valeur de votre paramètre cible.

L'exemple de code suivant remplace tous les paramètres dont la valeur est égale `c:\oozie\hive.py` à par la `s3://bucket-oozie/hive.py` valeur.

```
AddParameterMapping
  -treePath: 'ETL.OOZIE.Applications'
  -sourceValue: 'c:\oozie\hive.py'
```

```
-targetValue: 's3://bucket-oozie/hive.py'  
/
```

L'exemple de code suivant remplace tous les paramètres dont le nom est égal à `nameNode` à par la `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` valeur.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'  
/
```

L'exemple de code suivant remplace tous les paramètres dont le nom est égal à `nameNode` et la valeur est égale `hdfs://ip-55.eu-west-1.compute.internal:8020` à par la valeur du `targetValue` paramètre.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'  
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'  
/
```

Pour ajouter un nouveau paramètre dans vos fichiers cibles en plus d'un paramètre existant dans vos fichiers source, utilisez la `AddTargetParameter` commande. Cette commande utilise le même ensemble de paramètres que la `AddParameterMapping` commande.

L'exemple de code suivant ajoute le paramètre `clusterId` cible au lieu du `nameNode` paramètre.

```
AddTargetParameter  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'  
-targetParameter: 'clusterId'  
-targetValue: '1234567890abcdef0'  
/
```

Pour plus d'informations sur les `CreateFilter` commandes `AddServerMapping`, `etAddParameterMapping`, `AddTargetParameter`,,,,,, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

## Étape 4 : Créer un rapport d'évaluation

Avant de commencer la conversion, nous vous recommandons de créer un rapport d'évaluation. Ce rapport résume toutes les tâches de migration et détaille les actions qui apparaîtront au cours de la migration. Pour vous assurer que votre migration n'échoue pas, consultez ce rapport et abordez les mesures à prendre avant la migration. Pour de plus amples informations, veuillez consulter [Rapport d'évaluation](#).

Dans cette étape, vous devez utiliser la `CreateReport` commande. Cette commande utilise deux paramètres. Le premier paramètre décrit les objets source pour lesquels un rapport d'évaluation est AWS SCT créé. Pour ce faire, utilisez l'un des paramètres suivants : `filterName` ou `treePath`. Ce paramètre est obligatoire. Vous pouvez également ajouter un paramètre booléen facultatif `forceLoad`. Si vous définissez ce paramètre sur `true`, charge AWS SCT automatiquement tous les objets enfants pour l'objet source que vous spécifiez dans la `CreateReport` commande.

L'exemple de code suivant crée un rapport d'évaluation pour le `Applications` nœud de vos fichiers source Oozie.

```
CreateReport
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Vous pouvez ensuite enregistrer une copie du rapport d'évaluation au format PDF ou au format CSV (valeurs séparées par des virgules). Pour ce faire, utilisez la `SaveReportCSV` ou `SaveReportPDF` commande.

La `SaveReportPDF` commande enregistre une copie de votre rapport d'évaluation sous forme de fichier PDF. Cette commande utilise quatre paramètres. Le `file` paramètre est obligatoire, les autres paramètres sont facultatifs.

- `file`— le chemin d'accès au fichier PDF et son nom.
- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `namePath`: le chemin qui inclut uniquement les noms des objets cibles pour lesquels vous enregistrez une copie du rapport d'évaluation.



La `SaveReportCSV` commande enregistre votre rapport d'évaluation dans des fichiers CSV. Cette commande utilise quatre paramètres. Le `directory` paramètre est obligatoire, les autres paramètres sont facultatifs.

- `directory`— le chemin d'accès au dossier dans lequel AWS SCT sont enregistrés les fichiers CSV.
- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `namePath`: le chemin qui inclut uniquement les noms des objets cibles pour lesquels vous enregistrez une copie du rapport d'évaluation.

L'exemple de code suivant enregistre une copie du rapport d'évaluation dans le `c:\sct\ar.pdf` fichier.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

L'exemple de code suivant enregistre une copie du rapport d'évaluation sous forme de fichiers CSV dans le `c:\sct` dossier.

```
SaveReportCSV
-file:'c:\sct'
/
```

Pour plus d'informations sur les `SaveReportCSV` commandes `SaveReportPDF` et `CreateReport`, consultez la [référence de la AWS Schema Conversion Tool CLI](#).

## Étape 5 : Convertissez vos flux de travail Apache Oozie en AWS Step FunctionsAWS SCT

Après avoir configuré votre AWS SCT projet, convertissez votre code source et appliquez-le au AWS Cloud.

Au cours de cette étape, vous devez utiliser les `ApplyToTarget` commandes `Convert` `SaveOnS3ConfigureStateMachine`, et.

La `Migrate` commande fait migrer vos objets source vers le cluster cible. Cette commande utilise quatre paramètres. Assurez-vous de spécifier le `treePath` paramètre `filter` ou. Les autres paramètres sont facultatifs.

- `filter`: le nom du filtre que vous avez créé auparavant pour définir l'étendue des objets source à migrer.
- `namePath`— le chemin explicite vers un objet source spécifique.
- `treePath`— le chemin explicite vers les objets de votre base de données source pour lesquels vous enregistrez une copie du rapport d'évaluation.
- `forceLoad`— lorsqu'il est défini sur `true`, charge AWS SCT automatiquement les arborescences de métadonnées de la base de données lors de la migration. La valeur par défaut est `false`.

L'exemple de code suivant convertit les fichiers du Applications dossier de vos fichiers source Oozie.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

`SaveOnS3` Télécharge ensuite les définitions des machines d'état dans votre compartiment Amazon S3. Cette commande utilise le `treePath` paramètre. Pour exécuter cette commande, utilisez le dossier cible contenant les définitions des machines d'état comme valeur de ce paramètre.

Ce qui suit télécharge le `State machine definitions` dossier de votre objet `AWS_STEP_FUNCTIONS` cible dans le compartiment Amazon S3. AWS SCT utilise le compartiment Amazon S3 que vous avez stocké dans le profil AWS de service lors de l'[Prérequis](#) étape.

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

La `ConfigureStateMachine` commande configure les machines d'état. Cette commande utilise jusqu'à six paramètres. Assurez-vous de définir l'étendue cible à l'aide de l'un des trois premiers paramètres de la liste suivante.

- `filterName`— le nom du filtre pour vos objets cibles. Vous pouvez créer un filtre à l'aide de la `CreateFilter` commande.

- `treePath`— le chemin explicite vers vos objets cibles.
- `namePath`— le chemin explicite vers un objet cible spécifique.
- `iamRole`— le nom de ressource Amazon (ARN) du rôle IAM qui permet d'accéder à vos machines à étapes. Ce paramètre est obligatoire.

L'exemple de code suivant configure les machines à états définies à `AWS_STEP_FUNCTIONS` l'aide du rôle `role_name` IAM.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

La `ApplyToTarget` commande applique votre code converti au serveur cible. Pour exécuter cette commande, utilisez l'un des paramètres suivants : `filterNameTreePath`, ou `namePath` pour définir les objets cibles à appliquer.

L'exemple de code suivant applique la machine à `app_wp` états à AWS Step Functions.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Pour vous assurer que votre code converti produit les mêmes résultats que votre code source, vous pouvez utiliser le pack d' AWS SCT extension. Il s'agit d'un ensemble de AWS Lambda fonctions qui émulent vos fonctions Apache Oozie qui AWS Step Functions ne sont pas prises en charge. Pour installer ce pack d'extension, vous pouvez utiliser la `CreateLambdaExtPack` commande.

Cette commande utilise jusqu'à cinq paramètres. Assurez-vous que vous utilisez **Oozie2SF** `pourentPackId`. Dans ce cas, AWS SCT crée un pack d'extension pour les fonctions source d'Apache Oozie.

- `extPackId`— l'identifiant unique d'un ensemble de fonctions Lambda. Ce paramètre est obligatoire.
- `tempDirectory`— le chemin où AWS SCT peuvent être stockés les fichiers temporaires. Ce paramètre est obligatoire.
- `awsProfile`— le nom de votre AWS profil.

- `lambdaExecRoles`— la liste des Amazon Resource Names (ARNs) des rôles d'exécution à utiliser pour les fonctions Lambda.
- `createInvokeRoleFlag`— l'indicateur booléen qui indique s'il faut créer un rôle d'exécution pour. AWS Step Functions

Pour installer et utiliser le pack d'extension, assurez-vous de fournir les autorisations requises. Pour de plus amples informations, veuillez consulter [Autorisations d'utilisation AWS Lambda des fonctions du pack d'extension](#).

[Pour plus d'informations sur les `CreateLambdaExtPack` commandes,,  
`etConvert`,,  
`SaveOnS3`,  
`ConfigureStateMachine`,  
`ApplyToTarget`,,,,,,,,,,,,,,,,,,,,,,  
AWS Schema  
Conversion Tool](#)

## Exécution de votre script CLI

Une fois que vous avez terminé de modifier votre script AWS SCT CLI, enregistrez-le sous forme de fichier avec l'.sctsexension. Vous pouvez maintenant exécuter votre script depuis le app dossier correspondant à votre chemin AWS SCT d'installation. Pour cela, utilisez la commande suivante.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Dans l'exemple précédent, remplacez *script\_path* par le chemin d'accès à votre fichier par le script CLI. Pour plus d'informations sur l'exécution de scripts CLI dans AWS SCT, consultez [Mode script](#).

## Nœuds Apache Oozie AWS SCT pouvant être convertis en AWS Step Functions

Vous pouvez l'utiliser AWS SCT pour convertir les nœuds d'action Apache Oozie et les nœuds de flux de contrôle en. AWS Step Functions

Les nœuds d'action pris en charge sont les suivants :

- Action de la ruche
- Action Hive2
- Action Spark
- MapReduce Action de diffusion

- Action Java
- DistCp action
- Action porcine
- Action Sqoop
- Action FS
- Action de la coque

Les nœuds de flux de contrôle pris en charge sont les suivants :

- Lancer l'action
- Mettre fin à l'action
- Action de destruction
- Action décisionnelle
- Action de la fourche
- Rejoignez l'action

# Intégration AWS Database Migration Service avec AWS Schema Conversion Tool

## Utilisation d'un agent AWS SCT de réplication avec AWS DMS

Pour les migrations de bases de données de très grande envergure, vous pouvez utiliser un agent de AWS SCT réplication (`aws-schema-conversion-tool-dms-agent`) pour copier les données de votre base de données sur site vers Amazon S3 ou un appareil Edge. AWS Snowball Edge L'agent de réplication fonctionne conjointement avec AWS DMS et peut fonctionner en arrière-plan lorsqu' AWS SCT il est fermé.

Lorsque vous travaillez avec AWS Snowball Edge Edge, l' AWS SCT agent réplique les données sur l' AWS Snowball Edge appareil. L'appareil est ensuite envoyé AWS et les données sont chargées dans un compartiment Amazon S3. Pendant ce temps, l' AWS SCT agent continue de fonctionner. L'agent prend ensuite les données sur Amazon S3 et les copie vers le point de terminaison cible.

Pour de plus amples informations, veuillez consulter [Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool](#).

## Utilisation d'un agent d'extraction de AWS SCT données avec AWS DMS

Dans AWS SCT, vous trouverez un agent d'extraction de données (`aws-schema-conversion-tool-extractor`) qui facilite les migrations d'Apache Cassandra vers Amazon DynamoDB. Cassandra et DynamoDB sont des bases de données NoSQL, mais leur architecture système et leur représentation des données diffèrent. Vous pouvez utiliser des flux de travail basés sur des assistants AWS SCT pour automatiser le processus de Cassandra-to-DynamoDB migration. AWS SCT s'intègre à AWS Database Migration Service (AWS DMS) pour effectuer la migration proprement dite.

Pour de plus amples informations, veuillez consulter [Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool](#).

# Augmenter les niveaux de journalisation lors de l'utilisation AWS SCT avec AWS DMS

Vous pouvez augmenter les niveaux de journalisation lorsque vous utilisez AWS SCT avec AWS DMS, par exemple, si vous devez travailler avec le AWS Support.

Après l'installation AWS SCT et les pilotes requis, ouvrez l'application en choisissant l' AWS SCT icône. Si une notification de mise à jour s'affiche, vous pouvez choisir de procéder à la mise à jour avant ou après la fin de votre projet. Si une fenêtre de projet automatique s'ouvre, fermez-la et créez un projet manuellement.

Pour augmenter les niveaux de journalisation lors de l'utilisation AWS SCT avec AWS DMS

1. Dans le menu Paramètres, choisissez Paramètres globaux.
2. Dans la fenêtre des paramètres globaux, choisissez Logging.
3. Pour le mode Debug, choisissez True.
4. Dans la section Niveau du message, vous pouvez modifier les types de journaux suivants :
  - Général
  - Chargeur
  - Analyseur
  - Imprimante
  - Résolveur
  - Télémétrie
  - Convertisseur

Par défaut, tous les niveaux de message sont définis sur Info.

5. Choisissez un niveau de journalisation pour tous les types de messages que vous souhaitez modifier :
  - Trace (journalisation la plus détaillée)
  - Débogage
  - Infos
  - Avertissement
  - Erreur (journalisation la moins détaillée)

- Critique
  - Obligatoire
6. Choisissez Appliquer pour modifier les paramètres de votre projet.
  7. Cliquez sur OK pour fermer la fenêtre des paramètres généraux.



# Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool

Vous pouvez utiliser un AWS SCT agent pour extraire les données de votre entrepôt de données sur site et les migrer vers Amazon Redshift. L'agent extrait vos données et les télécharge sur Amazon S3 ou, pour les migrations à grande échelle, sur un appareil AWS Snowball Edge Edge. Vous pouvez ensuite utiliser un AWS SCT agent pour copier les données sur Amazon Redshift.

Vous pouvez également utiliser AWS Database Migration Service (AWS DMS) pour migrer des données vers Amazon Redshift. L'avantage de AWS DMS est la prise en charge de la réplication continue (capture des données de modification). Toutefois, pour accélérer la migration des données, utilisez plusieurs AWS SCT agents en parallèle. Selon nos tests, les AWS SCT agents migrent les données plus rapidement que AWS DMS de 15 à 35 %. La différence de vitesse est due à la compression des données, à la prise en charge de la migration des partitions de table en parallèle et aux différents paramètres de configuration. Pour plus d'informations, consultez [Utilisation d'une base de données Amazon Redshift comme cible pour AWS Database Migration Service](#).

Amazon S3 est un service de stockage et d'extraction. Pour stocker un objet dans Amazon S3, vous chargez le fichier correspondant dans un compartiment Amazon S3. Lorsque vous chargez un fichier, vous pouvez définir les autorisations sur l'objet, ainsi que celles liées aux métadonnées.

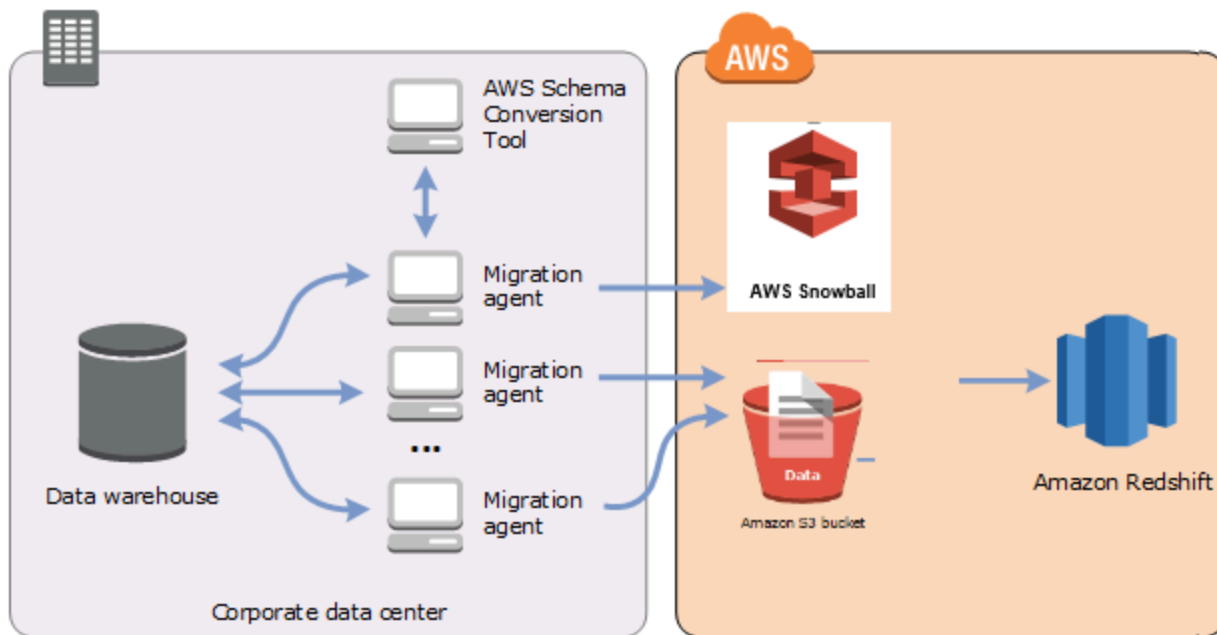
## Migrations à grande échelle

Les migrations de données à grande échelle peuvent inclure de nombreux téraoctets d'informations et peuvent être ralenties par les performances du réseau et par la quantité de données à déplacer. AWS Snowball Edge Edge est un AWS service que vous pouvez utiliser pour transférer des données vers le cloud à faster-than-network grande vitesse à l'aide d'une appliance AWS appartenant à votre propriétaire. Un appareil AWS Snowball Edge Edge peut contenir jusqu'à 100 To de données. Il utilise un cryptage 256 bits et un module TPM (Trusted Platform Module) conforme aux normes du secteur pour garantir à la fois la sécurité et l'intégralité chain-of-custody de vos données. AWS SCT fonctionne avec les appareils AWS Snowball Edge Edge.

Lorsque vous utilisez AWS SCT un appareil AWS Snowball Edge Edge, vous migrez vos données en deux étapes. Tout d'abord, AWS SCT vous traite les données localement, puis vous les déplacez vers le périphérique AWS Snowball Edge Edge. Vous envoyez ensuite l'appareil à AWS l'aide du processus AWS Snowball Edge Edge, puis vous chargez AWS automatiquement les données dans un compartiment Amazon S3. Ensuite, lorsque les données sont disponibles sur Amazon S3, vous

pouvez les migrer AWS SCT vers Amazon Redshift. Les agents d'extraction de données peuvent travailler en arrière-plan lorsqu' AWS SCT il est fermé.

Le schéma suivant illustre le scénario pris en charge.



Les agents d'extraction de données sont actuellement pris en charge pour les entrepôts de données source suivants :

- Analyses Azure Synapse
- BigQuery
- Base de données Greenplum (version 4.3)
- Microsoft SQL Server (version 2008 et supérieure)
- Netezza (version 7.0.3 et supérieure)
- Oracle (version 10 et supérieure)
- Flocon de neige (version 3)
- Teradata (version 13 et supérieure)
- Vertica (version 7.2.2 et supérieure)

Vous pouvez vous connecter aux points de terminaison FIPS pour Amazon Redshift si vous devez vous conformer aux exigences de sécurité de la Federal Information Processing Standard (FIPS). Les points de terminaison FIPS sont disponibles dans les régions suivantes : AWS

- Région de l'Est des États-Unis (Virginie du Nord) ([redshift-fips.us-east-1.amazonaws.com](https://redshift-fips.us-east-1.amazonaws.com))
- Région de l'est des États-Unis (Ohio) ([redshift-fips.us-east-2.amazonaws.com](https://redshift-fips.us-east-2.amazonaws.com))
- Région ouest des États-Unis (Californie du Nord) ([redshift-fips.us-west-1.amazonaws.com](https://redshift-fips.us-west-1.amazonaws.com))
- Région ouest des États-Unis (Oregon) ([redshift-fips.us-west-2.amazonaws.com](https://redshift-fips.us-west-2.amazonaws.com))

Reportez-vous aux informations contenues dans les rubriques suivantes pour savoir comment utiliser les agents d'extraction de données.

## Rubriques

- [Conditions préalables à l'utilisation d'agents d'extraction de données](#)
- [Installation d'agents d'extraction](#)
- [Configuration des agents d'extraction](#)
- [Enregistrement des agents d'extraction auprès du AWS Schema Conversion Tool](#)
- [Masquer et récupérer des informations pour un AWS SCT agent](#)
- [Création de règles de migration de données dans AWS SCT](#)
- [Modification des paramètres de l'extracteur et de la copie à partir des paramètres du projet](#)
- [Tri des données avant la migration à l'aide de AWS SCT](#)
- [Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données](#)
- [Exportation et importation d'une tâche d'extraction de AWS SCT données](#)
- [Extraction de données à l'aide d'un appareil AWS Snowball Edge Edge](#)
- [Sortie de la tâche d'extraction de données](#)
- [Utilisation du partitionnement virtuel avec AWS Schema Conversion Tool](#)
- [Utilisation du partitionnement natif](#)
- [Migration LOBs vers Amazon Redshift](#)
- [Bonnes pratiques et résolution des problèmes pour les agents d'extraction de données](#)

# Conditions préalables à l'utilisation d'agents d'extraction de données

Avant de travailler avec des agents d'extraction de données, ajoutez les autorisations requises pour Amazon Redshift en tant que cible pour votre utilisateur Amazon Redshift. Pour de plus amples informations, veuillez consulter [Autorisations pour Amazon Redshift en tant que cible](#).

Stockez ensuite les informations de votre compartiment Amazon S3 et configurez votre système de confiance SSL (Secure Sockets Layer) et votre stockage de clés.

## Paramètres Amazon S3

Une fois que vos agents ont extrait vos données, ils les téléchargent dans votre compartiment Amazon S3. Avant de continuer, vous devez fournir les informations d'identification pour vous connecter à votre AWS compte et à votre compartiment Amazon S3. Vous stockez vos informations d'identification et de compartiment dans un profil dans les paramètres globaux de l'application, puis vous associez le profil à votre AWS SCT projet. Si nécessaire, choisissez Paramètres généraux pour créer un nouveau profil. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).

Pour migrer des données vers votre base de données Amazon Redshift cible, l'agent d'extraction de AWS SCT données doit être autorisé à accéder au compartiment Amazon S3 en votre nom. Pour fournir cette autorisation, créez un utilisateur AWS Identity and Access Management (IAM) avec la politique suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/*",
        "arn:aws:s3:::bucket_name"
      ]
    }
  ]
}
```

```
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation"
    ],
    "Resource": [
      "arn:aws:s3:::bucket_name"
    ],
    "Effect": "Allow"
  },
  {
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  },
  {
    "Action": [
      "iam:GetUser"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:user/DataExtractionAgentName"
    ],
    "Effect": "Allow"
  }
]
}
```

Dans l'exemple précédent, remplacez *bucket\_name* par le nom de votre compartiment Amazon S3. Remplacez ensuite *111122223333:user/DataExtractionAgentName* par le nom de votre utilisateur IAM.

## Assumer des rôles IAM

Pour plus de sécurité, vous pouvez utiliser des rôles AWS Identity and Access Management (IAM) pour accéder à votre compartiment Amazon S3. Pour ce faire, créez un utilisateur IAM pour vos agents d'extraction de données sans aucune autorisation. Créez ensuite un rôle IAM qui autorise l'accès à Amazon S3 et spécifiez la liste des services et des utilisateurs qui peuvent assumer ce rôle. Pour plus d'informations, consultez [Rôles IAM](#) dans le Guide de l'utilisateur IAM.

## Pour configurer les rôles IAM afin d'accéder à votre compartiment Amazon S3

1. Créez un nouvel utilisateur IAM. Pour les informations d'identification de l'utilisateur, choisissez le type d'accès programmatique.
2. Configurez l'environnement hôte afin que votre agent d'extraction de données puisse assumer le rôle qui lui est AWS SCT attribué. Assurez-vous que l'utilisateur que vous avez configuré à l'étape précédente autorise les agents d'extraction de données à utiliser la chaîne de fournisseurs d'informations d'identification. Pour plus d'informations, consultez la section [Utilisation des informations d'identification](#) dans le Guide du AWS SDK pour Java développeur.
3. Créez un nouveau rôle IAM qui a accès à votre compartiment Amazon S3.
4. Modifiez la section de confiance de ce rôle pour autoriser l'utilisateur que vous avez créé auparavant à assumer le rôle. Dans l'exemple suivant, remplacez *111122223333:user/DataExtractionAgentName* par le nom de votre utilisateur.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
  },
  "Action": "sts:AssumeRole"
}
```

5. Modifiez la section de confiance de ce rôle en faisant confiance `redshift.amazonaws.com` pour assumer le rôle.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

6. Associez ce rôle à votre cluster Amazon Redshift.

Vous pouvez désormais exécuter votre agent d'extraction de données dans AWS SCT.

Lorsque vous utilisez l'acceptation du rôle IAM, la migration des données fonctionne de la manière suivante. L'agent d'extraction de données démarre et obtient les informations d'identification de l'utilisateur à l'aide de la chaîne de fournisseurs d'informations d'identification. Ensuite, vous créez une tâche de migration de données dans AWS SCT, puis vous spécifiez le rôle IAM que les agents d'extraction de données doivent assumer, puis vous lancez la tâche. AWS Security Token Service (AWS STS) génère des informations d'identification temporaires pour accéder à Amazon S3. L'agent d'extraction de données utilise ces informations d'identification pour charger des données sur Amazon S3.

AWS SCT Fournit ensuite à Amazon Redshift le rôle IAM. À son tour, Amazon Redshift obtient de nouvelles informations d'identification temporaires AWS STS pour accéder à Amazon S3. Amazon Redshift utilise ces informations d'identification pour copier les données d'Amazon S3 vers votre table Amazon Redshift.

## Paramètres de sécurité

Les agents AWS Schema Conversion Tool et les agents d'extraction peuvent communiquer via le protocole SSL (Secure Sockets Layer). Pour activer SSL, configurez un référentiel d'approbations et un magasin de clés.

Pour configurer la communication avec votre agent d'extraction

1. Démarrez le AWS Schema Conversion Tool.
2. Ouvrez le menu Paramètres, puis choisissez Paramètres globaux. La boîte de dialogue Global settings s'affiche.
3. Choisissez Security (Sécurité).
4. Choisissez Generate trust and key store, ou sélectionnez Select existing trust store.

Si vous choisissez Generate trust and key store, vous spécifiez ensuite le nom et le mot de passe pour les trust et key stores, ainsi que le chemin d'accès à l'emplacement des fichiers générés. Vous utiliserez ces fichiers au cours d'étapes ultérieures.

Si vous choisissez Sélectionner un magasin de confiance existant, vous devez ensuite spécifier le mot de passe et le nom de fichier pour le magasin de confiance et de clés. Vous utiliserez ces fichiers au cours d'étapes ultérieures.

5. Après avoir spécifié le trust store et le key store, cliquez sur OK pour fermer la boîte de dialogue des paramètres globaux.

## Configuration de l'environnement pour les agents d'extraction de données

Vous pouvez installer plusieurs agents d'extraction de données sur un même hôte. Toutefois, nous vous recommandons d'exécuter un agent d'extraction de données sur un hôte.

Pour exécuter votre agent d'extraction de données, assurez-vous d'utiliser un hôte doté d'au moins quatre V CPUs et de 32 Go de mémoire. Définissez également la mémoire minimale disponible AWS SCT à au moins quatre Go. Pour de plus amples informations, veuillez consulter [Configuration de mémoire supplémentaire](#).

La configuration optimale et le nombre d'hôtes d'agents dépendent de la situation spécifique de chaque client. Assurez-vous de prendre en compte des facteurs tels que la quantité de données à migrer, la bande passante du réseau, le temps nécessaire pour extraire les données, etc. Vous pouvez d'abord effectuer une preuve de concept (PoC), puis configurer vos agents d'extraction de données et vos hôtes en fonction des résultats de cette validation.

## Installation d'agents d'extraction

Nous vous recommandons d'installer plusieurs agents d'extraction et d'effectuer ces installations sur des ordinateurs autres que l'ordinateur qui exécute AWS Schema Conversion Tool.

Les agents d'extraction sont actuellement pris en charge sur les systèmes d'exploitation suivants :

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (version 14.04 et supérieure)

Utilisez la procédure suivante pour installer des agents d'extraction. Répétez cette procédure pour chaque ordinateur sur lequel vous souhaitez installer un agent d'extraction.

Pour installer un agent d'extraction

1. Si vous n'avez pas encore téléchargé le fichier AWS SCT d'installation, suivez les instructions [Installation et configuration AWS Schema Conversion Tool](#) pour le télécharger. Le fichier .zip qui contient le fichier AWS SCT d'installation contient également le fichier d'installation de l'agent d'extraction.



2. Téléchargez et installez la dernière version d'Amazon Corretto 11. Pour plus d'informations, consultez la section [Téléchargements pour Amazon Corretto 11 dans le guide de l'utilisateur d'Amazon Corretto 11](#).
3. Recherchez le fichier du programme d'installation de l'agent d'extraction dans un sous-dossier nommé agents. Le fichier correct pour chaque système d'exploitation sur lequel vous souhaitez installer l'agent d'extraction est illustré ci-après.

Système d'exploitation	Nom de fichier
Microsoft Windows	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .msi
RHEL	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm
Ubuntu Linux	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb

4. Installez l'agent d'extraction sur un autre ordinateur en copiant le fichier d'installation sur le nouvel ordinateur.
5. Exécutez le fichier du programme d'installation. Suivez les instructions ci-dessous correspondant à votre système d'exploitation.

Système d'exploitation	Instructions d'installation
Microsoft Windows	Double-cliquez sur le fichier pour exécuter le programme d'installation.
RHEL	Exécutez les commandes suivantes dans le dossier dans lequel vous avez téléchargé ou déplacé le fichier. <pre> sudo rpm -ivh aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm sudo ./sct-extractor-setup.sh --config </pre>

Système d'exploitation	Instructions d'installation
Ubuntu Linux	<p>Exécutez les commandes suivantes dans le dossier dans lequel vous avez téléchargé ou déplacé le fichier.</p> <pre data-bbox="521 394 1507 548">sudo dpkg -i aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

6. Choisissez Suivant, acceptez le contrat de licence, puis cliquez sur Suivant.
7. Entrez le chemin d'installation de l'agent d'extraction de AWS SCT données, puis choisissez Next.
8. Choisissez Installer pour installer votre agent d'extraction de données.

AWS SCT installe votre agent d'extraction de données. Pour terminer l'installation, configurez votre agent d'extraction de données. AWS SCT lance automatiquement le programme de configuration. Pour de plus amples informations, veuillez consulter [Configuration des agents d'extraction](#).

9. Choisissez Terminer pour fermer l'assistant d'installation après avoir configuré votre agent d'extraction de données.

## Configuration des agents d'extraction

Utilisez la procédure suivante pour configurer des agents d'extraction. Répétez cette procédure sur chaque ordinateur sur lequel un agent d'extraction est installé.

Pour configurer votre agent d'extraction

1. Lancez le programme de configuration :
  - Sous Windows, AWS SCT lance automatiquement le programme de configuration lors de l'installation d'un agent d'extraction de données.

Au besoin, vous pouvez lancer le programme d'installation manuellement. Pour ce faire, exécutez le `ConfigAgent.bat` fichier sous Windows. Ce fichier se trouve dans le dossier où vous avez installé l'agent.

- Dans RHEL et Ubuntu, exécutez le `sct-extractor-setup.sh` fichier à partir de l'emplacement où vous avez installé l'agent.

Le programme de configuration vous invite à fournir des informations. Pour chaque invite, une valeur par défaut apparaît.

2. Acceptez la valeur par défaut à chaque invite ou entrez une nouvelle valeur.

Spécifiez les informations suivantes :

- Pour Port d'écoute, entrez le numéro de port sur lequel l'agent écoute.
- Pour Ajouter un fournisseur source, entrez oui, puis entrez votre plateforme d'entrepôt de données source.
- Pour le pilote JDBC, entrez l'emplacement où vous avez installé les pilotes JDBC.
- Dans le champ Dossier de travail, entrez le chemin dans lequel l'agent d'extraction de AWS SCT données stockera les données extraites. Le dossier de travail peut figurer sur un autre ordinateur que celui où est installé l'agent et un même dossier de travail peut être partagé par plusieurs agents installés sur différents ordinateurs.
- Pour Activer la communication SSL, entrez oui.
- Pour Key store, entrez l'emplacement du fichier key store.
- Dans le champ Mot de passe du magasin de clés, entrez le mot de passe du magasin de clés.
- Pour Activer l'authentification SSL du client, entrez oui.
- Pour Trust store, entrez l'emplacement du fichier Trust Store.
- Dans le champ Mot de passe du Trust Store, entrez le mot de passe du Trust Store.

Le programme de configuration met à jour le fichier de paramètres de l'agent d'extraction. Le fichier de paramètres se nomme `settings.properties` et se trouve à l'emplacement où vous avez installé l'agent d'extraction.

Vous trouverez ci-dessous un exemple de fichier de paramètres.

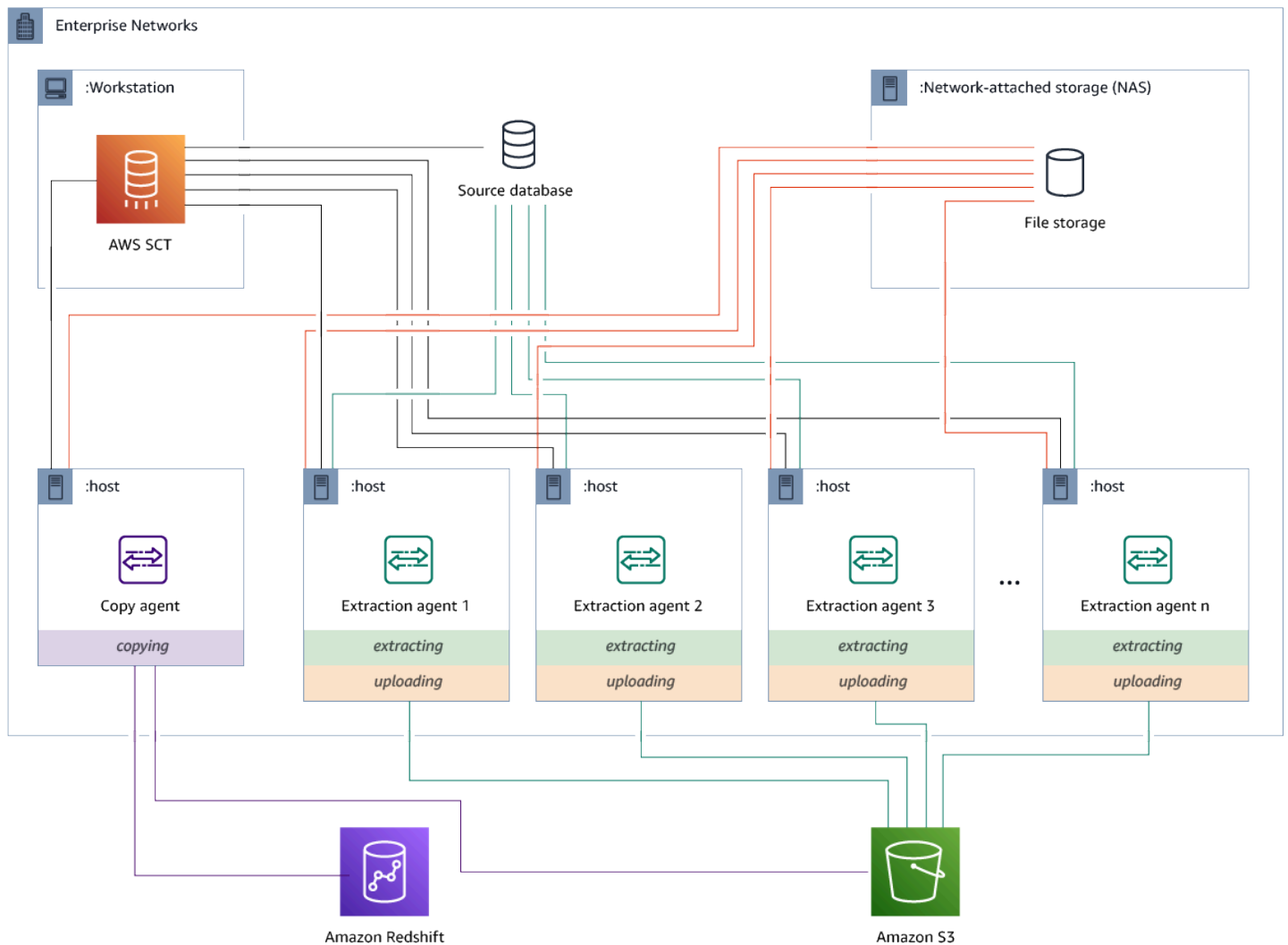
```
$ cat settings.properties
#extractor.start.fetch.size=20000
#extractor.out.file.size=10485760
#extractor.source.connection.pool.size=20
#extractor.source.connection.pool.min.evictable.idle.time.millis=30000
#extractor.extracting.thread.pool.size=10
```

```
vendor=TERADATA
driver.jars=/usr/share/lib/jdbc/terajdbc4.jar
port=8192
redshift.driver.jars=/usr/share/lib/jdbc/RedshiftJDBC42-1.2.43.1067.jar
working.folder=/data/sct
extractor.private.folder=/home/ubuntu
ssl.option=OFF
```

Pour modifier les paramètres de configuration, vous pouvez modifier le `settings.properties` fichier à l'aide d'un éditeur de texte ou réexécuter la configuration de l'agent.

## Installation et configuration d'agents d'extraction avec des agents de copie dédiés

Vous pouvez installer des agents d'extraction dans une configuration dotée d'un stockage partagé et d'un agent de copie dédié. Le schéma suivant illustre ce scénario.



Cette configuration peut être utile lorsqu'un serveur de base de données source prend en charge jusqu'à 120 connexions et que votre réseau dispose d'une capacité de stockage suffisante. Suivez la procédure ci-dessous pour configurer les agents d'extraction dotés d'un agent de copie dédié.

Pour installer et configurer des agents d'extraction et un agent de copie dédié

1. Assurez-vous que le répertoire de travail de tous les agents d'extraction utilise le même dossier sur le stockage partagé.
2. Installez les agents d'extraction en suivant les étapes décrites dans [Installation d'agents d'extraction](#).
3. Configurez les agents d'extraction en suivant les étapes décrites [Configuration des agents d'extraction](#), mais spécifiez uniquement le pilote JDBC source.

- Configurez un agent de copie dédié en suivant les étapes décrites [Configuration des agents d'extraction](#), mais spécifiez uniquement un pilote Amazon Redshift JDBC.

## Agents d'extraction de départ

Utilisez la procédure suivante pour démarrer des agents d'extraction. Répétez cette procédure sur chaque ordinateur sur lequel un agent d'extraction est installé.

Les agents d'extraction agissent en tant qu'écouteurs. Lorsque vous démarrez un agent avec cette procédure, celui-ci commence à écouter les instructions. Vous enverrez aux agents des instructions pour extraire des données de votre entrepôt de données dans une section ultérieure.

Pour démarrer votre agent d'extraction

- Sur l'ordinateur où l'agent d'extraction a été installé, exécutez la commande indiquée ci-après pour votre système d'exploitation.

Système d'exploitation	Commande de démarrage
Microsoft Windows	Double-cliquez sur le fichier de commandes <code>StartAgent.bat</code> .
RHEL	Exécutez la commande suivante dans le chemin d'accès au dossier où vous avez installé l'agent :  <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	Exécutez la commande suivante dans le chemin d'accès au dossier où vous avez installé l'agent. Utilisez la commande appropriée pour votre version d'Ubuntu.  Ubuntu 14.04: <code>sudo initctl <i>start</i> sct-extractor</code>  Ubuntu 15.04 et versions ultérieures : <code>sudo systemctl <i>start</i> sct-extractor</code>

Pour vérifier le statut de l'agent, exécutez la même commande en remplaçant `start` par `status`.

Pour arrêter un agent, exécutez la même commande en remplaçant `start` par `stop`.

# Enregistrement des agents d'extraction auprès du AWS Schema Conversion Tool

Vous gérez vos agents d'extraction en utilisant AWS SCT. Les agents d'extraction agissent en tant qu'écouteurs. Lorsqu'ils reçoivent des instructions de leur part AWS SCT, ils extraient les données de votre entrepôt de données.

Utilisez la procédure suivante pour enregistrer les agents d'extraction dans votre AWS SCT projet.

Pour enregistrer un agent d'extraction

1. Démarrez AWS Schema Conversion Tool et ouvrez un projet.
2. Ouvrez le menu Affichage, puis choisissez Mode migration des données (autre). L'onglet Agents s'affiche. Si vous avez déjà enregistré des agents, AWS SCT affichez-les dans une grille en haut de l'onglet.
3. Choisissez Register (S'inscrire).

Une fois que vous avez enregistré un agent dans un AWS SCT projet, vous ne pouvez pas enregistrer le même agent dans un autre projet. Si vous n'utilisez plus d'agent dans un AWS SCT projet, vous pouvez annuler son enregistrement. Vous pouvez ensuite l'enregistrer dans un autre projet.

4. Choisissez l'agent de données Redshift, puis cliquez sur OK.
5. Entrez vos informations dans l'onglet Connexion de la boîte de dialogue :
  - a. Dans Description, entrez une description de l'agent.
  - b. Dans Nom d'hôte, entrez le nom d'hôte ou l'adresse IP de l'ordinateur de l'agent.
  - c. Pour Port, entrez le numéro de port sur lequel l'agent écoute.
  - d. Choisissez Enregistrer pour enregistrer l'agent auprès de votre AWS SCT projet.
6. Répétez les étapes précédentes pour enregistrer plusieurs agents auprès de votre projet AWS SCT .

## Masquer et récupérer des informations pour un AWS SCT agent

Un AWS SCT agent chiffre une quantité importante d'informations, par exemple les mots de passe des banques de données de confiance des utilisateurs, les comptes de base de données, les

informations relatives aux comptes et autres éléments similaires. AWS Pour ce faire, il utilise un fichier spécial nommé `seed.dat`. Par défaut, l'agent crée ce fichier dans le dossier de travail de l'utilisateur qui configure l'agent pour la première fois.

Dans la mesure où différents utilisateurs peuvent configurer et exécuter l'agent, le chemin d'accès à `seed.dat` est stocké dans le paramètre `{extractor.private.folder}` du fichier `settings.properties`. Lorsque l'agent démarre, il peut utiliser ce chemin pour trouver le fichier `seed.dat` afin d'accéder aux informations du référentiel d'approbations et du magasin de clés de la base de données qu'il utilise.

Il se peut que vous ayez besoin de récupérer les mots de passe stockés par un agent dans les cas suivants :

- Si l'utilisateur perd le `seed.dat` fichier et que l'emplacement et le port de l' AWS SCT agent n'ont pas changé.
- Si l'utilisateur perd le `seed.dat` fichier et que l'emplacement et le port de l' AWS SCT agent ont changé. Dans ce cas, la modification se produit généralement parce que l'agent a été migré vers un autre hôte ou port et que les informations contenues dans le fichier `seed.dat` ne sont plus valides.

Dans de tels cas, si un agent est démarré sans SSL, il démarre, puis accède au stockage de l'agent créé précédemment. Il passe ensuite à l'état `Waiting for recovery` (En attente de récupération).

Cependant, dans ces cas, si un agent est démarré avec SSL, vous ne pouvez pas le redémarrer. En effet, l'agent ne peut pas déchiffrer les mots de passe pour accéder aux certificats stockés dans le fichier `settings.properties`. Dans ce type de démarrage, l'agent ne parvient pas à démarrer. Une erreur similaire à la suivante est écrite dans le journal : « `The agent could not start with SSL mode enabled. Please reconfigure the agent. Reason: The password for keystore is incorrect.` » (L'agent n'a pas pu démarrer avec le mode SSL activé. Veuillez reconfigurer l'agent. Raison : le mot de passe du keystore est incorrect.)

Pour corriger ce problème, créez un nouvel agent et configurez-le afin qu'il utilise les mots de passe existants pour accéder aux certificats SSL. Pour cela, procédez comme suit.

Après avoir effectué cette procédure, l'agent doit s'exécuter et passer à l'état `En attente de restauration`. AWS SCT envoie automatiquement les mots de passe nécessaires à un agent en état `En attente de restauration`. Lorsque l'agent dispose des mots de passe, il redémarre toutes les tâches. Aucune autre action de l'utilisateur n'est requise sur le AWS SCT côté.



Pour reconfigurer l'agent et restaurer les mots de passe afin d'accéder aux certificats SSL

1. Installez un nouvel AWS SCT agent et exécutez la configuration.
2. Modifiez la propriété `agent.name` dans le fichier `instance.properties` pour indiquer le nom de l'agent pour lequel le stockage a été créé, afin que le nouvel agent fonctionne avec le stockage de l'agent existant.

Le fichier `instance.properties` est stocké dans le dossier privé de l'agent, qui est nommé en utilisant la convention suivante : `{output.folder}\dmt\{hostName}_{portNumber}\`.

3. Remplacez le nom de `{output.folder}` par celui du dossier de sortie de l'agent précédent.

À ce stade, AWS SCT essaie toujours d'accéder à l'ancien extracteur sur l'ancien hôte et le port. Par conséquent, l'extracteur inaccessible est à l'état FAILED (ÉCHEC). Vous pouvez ensuite modifier l'hôte et le port.

4. Modifiez l'hôte et/ou le port de l'ancien agent à l'aide de la commande `Modify` pour rediriger le flux de demandes vers le nouvel agent.

Quand AWS SCT peut envoyer un ping au nouvel agent, AWS SCT reçoit le statut `En attente de restauration` de la part de l'agent. AWS SCT récupère ensuite automatiquement les mots de passe de l'agent.

Chaque agent qui fonctionne avec le stockage de l'agent met à jour un fichier spécial appelé `storage.lock` et situé dans `{output.folder}\{agentName}\storage\`. Ce fichier contient l'ID réseau de l'agent et le délai avant le verrouillage du stockage. Lorsque l'agent fonctionne avec le stockage de l'agent, il met à jour le fichier `storage.lock` et étend le bail du stockage de 10 minutes toutes les 5 minutes. Aucune autre instance ne peut fonctionner avec le stockage de cet agent avant l'expiration du bail.

## Création de règles de migration de données dans AWS SCT

Avant d'extraire vos données avec le AWS Schema Conversion Tool, vous pouvez configurer des filtres qui réduisent la quantité de données que vous extrayez. Vous pouvez créer des règles de migration des données en utilisant `WHERE` des clauses pour réduire le nombre de données que vous extrayez. Par exemple, vous pouvez écrire une clause `WHERE` qui sélectionne les données à partir d'une seule table.

Vous pouvez créer des règles de migration des données et enregistrer les filtres dans le cadre de votre projet. Votre projet étant ouvert, utilisez la procédure suivante pour créer des règles de migration de données.

Pour créer des règles de migration des données

1. Ouvrez le menu Affichage, puis choisissez Mode migration des données (autre).
2. Choisissez Règles de migration des données, puis sélectionnez Ajouter une nouvelle règle.
3. Configurez votre règle de migration des données :
  - a. Dans Nom, entrez le nom de votre règle de migration de données.
  - b. Pour Where schema name is like, entrez un filtre à appliquer aux schémas. Dans ce filtre, une clause WHERE est évaluée à l'aide d'une clause LIKE. Pour choisir un schéma, entrez un nom de schéma exact. Pour choisir plusieurs schémas, utilisez le caractère « % » comme caractère générique pour correspondre au nombre quelconque de caractères du nom du schéma.
  - c. Pour un nom de table, par exemple, entrez un filtre à appliquer aux tables. Dans ce filtre, une clause WHERE est évaluée à l'aide d'une clause LIKE. Pour choisir une table, entrez un nom exact. Pour sélectionner plusieurs tables, utilisez le caractère « % » comme caractère générique pour faire correspondre le nombre de caractères que vous souhaitez au nom de la table.
  - d. Pour la clause Where, entrez une WHERE clause pour filtrer les données.
4. Une fois que vous avez configuré votre filtre, choisissez Save pour enregistrer votre filtre ou Cancel pour annuler vos modifications.
5. Une fois que vous avez terminé d'ajouter, de modifier et de supprimer des filtres, choisissez Enregistrer tout pour enregistrer toutes vos modifications.

Pour désactiver un filtre sans le supprimer, vous pouvez utiliser l'icône bascule. Pour dupliquer un filtre existant, utilisez l'icône de copie. Pour supprimer un filtre existant, utilisez l'icône de suppression. Pour enregistrer les modifications que vous apportez à vos filtres, choisissez Enregistrer tout.

## Modification des paramètres de l'extracteur et de la copie à partir des paramètres du projet

Dans la fenêtre des paramètres du projet AWS SCT, vous pouvez choisir les paramètres des agents d'extraction de données et de la commande Amazon RedshiftCOPY.

Pour choisir ces paramètres, choisissez Paramètres, Paramètres du projet, puis Migration des données. Ici, vous pouvez modifier les paramètres d'extraction, les paramètres Amazon S3 et les paramètres de copie.

Suivez les instructions du tableau suivant pour fournir les informations relatives aux paramètres d'extraction.

Pour ce paramètre	Faites ceci
Format de compression	Spécifiez le format de compression des fichiers d'entrée. Choisissez l'une des options suivantes : GZIP BZIP2, ZSTD ou Aucune compression.
Caractère délimiteur	Spécifiez le caractère ASCII qui sépare les champs dans les fichiers d'entrée. Les caractères non imprimables ne sont pas pris en charge.
Valeur NULL sous forme de chaîne	Activez cette option si vos données incluent un terminateur nul. Si cette option est désactivée, la COPY commande Amazon Redshift considère la valeur null comme la fin de l'enregistrement et met fin au processus de chargement.
Stratégie de tri	Utilisez le tri pour relancer l'extraction à partir du point d'échec. Choisissez l'une des stratégies de tri suivantes : Utiliser le tri après le premier échec (recommandé), Utiliser le tri si possible ou Ne jamais utiliser le tri. Pour de plus amples informations, veuillez consulter <a href="#">the section called "Tri des données"</a> .
Schéma de température source	Entrez le nom du schéma dans la base de données source, où l'agent d'extraction peut créer les objets temporaires.

Pour ce paramètre	Faites ceci
Taille du fichier de sortie (en Mo)	Entrez la taille, en Mo, des fichiers chargés sur Amazon S3.
Taille du fichier Snowball Out (en Mo)	Entrez la taille, en Mo, des fichiers téléchargés vers AWS Snowball Edge. Les fichiers peuvent avoir une taille de 1 à 1 000 Mo.
Utilisez le partitionnement automatique. Pour Greenplum et Netezza, entrez la taille minimale des tables prises en charge (en mégaoctets)	Activez cette option pour utiliser le partitionnement des tables, puis entrez la taille des tables à partitionner pour les bases de données sources Greenplum et Netezza. Pour les migrations d'Oracle vers Amazon Redshift, vous pouvez laisser ce champ vide car cela AWS SCT crée des sous-tâches pour toutes les tables partitionnées.
Extrait LOBs	Activez cette option pour extraire des objets volumineux (LOBs) de votre base de données source. LOBs inclure BLOBs CLOBs, NCLOBs,, des fichiers XML, etc. Pour chaque LOB, les agents AWS SCT d'extraction créent un fichier de données.
LOBs Dossier de compartiment Amazon S3	Entrez l'emplacement des agents AWS SCT d'extraction à stocker LOBs.
Appliquer RTRIM aux colonnes de chaînes	Activez cette option pour supprimer un jeu de caractères spécifié à la fin des chaînes extraites.
Conservez les fichiers localement après leur téléchargement sur Amazon S3	Activez cette option pour conserver les fichiers sur votre machine locale une fois que les agents d'extraction de données les ont chargés sur Amazon S3.

Suivez les instructions du tableau suivant pour fournir les informations relatives aux paramètres Amazon S3.

Pour ce paramètre	Faites ceci
Utiliser un proxy	Activez cette option pour utiliser un serveur proxy afin de télécharger des données sur Amazon S3. Choisissez ensuite le protocole de transfert de données, entrez le nom d'hôte, le port, le nom d'utilisateur et le mot de passe.
Type de point de terminaison	Choisissez FIPS pour utiliser le point de terminaison FIPS (Federal Information Processing Standard). Choisissez VPCE pour utiliser le point de terminaison du cloud privé virtuel (VPC). Ensuite, pour le point de terminaison VPC, entrez le système de nom de domaine (DNS) de votre point de terminaison VPC.
Conservez les fichiers sur Amazon S3 après les avoir copiés sur Amazon Redshift	Activez cette option pour conserver les fichiers extraits sur Amazon S3 après les avoir copiés sur Amazon Redshift.

Suivez les instructions du tableau suivant pour fournir les informations relatives aux paramètres de copie.

Pour ce paramètre	Faites ceci
Nombre d'erreurs maximal	Entrez le nombre d'erreurs de chargement. Lorsque l'opération atteint cette limite, les agents d'extraction de AWS SCT données mettent fin au processus de chargement des données. La valeur par défaut est 0, ce qui signifie que les AWS SCT agents d'extraction de données continuent le chargement des données indépendamment des défaillances.
Remplace les caractères UTF-8 non valides	Activez cette option pour remplacer les caractères UTF-8 non valides par le caractère spécifié et poursuivre l'opération de chargement des données.
Utiliser le blanc comme valeur nulle	Activez cette option pour charger les champs vides composés d'espaces blancs sous la forme de valeurs nulles.

Pour ce paramètre	Faites ceci
Utiliser vide comme valeur nulle	Activez cette option pour charger des champs vides CHAR et VARCHAR des champs nuls.
Tronquer les colonnes	Activez cette option pour tronquer les données en colonnes afin de les adapter à la spécification du type de données.
Compression automatique	Activez cette option pour appliquer le codage de compression lors d'une opération de copie.
Actualisation automatique des statistiques	Activez cette option pour actualiser les statistiques à la fin d'une opération de copie.
Vérifier le fichier avant le chargement	Activez cette option pour valider les fichiers de données avant de les charger sur Amazon Redshift.

## Tri des données avant la migration à l'aide de AWS SCT

Le tri de vos données avant la migration AWS SCT présente certains avantages. Si vous trie d'abord les données, vous AWS SCT pouvez redémarrer l'agent d'extraction au dernier point enregistré après un échec. De plus, si vous migrez des données vers Amazon Redshift et que vous les trie d'abord AWS SCT , vous pouvez les insérer plus rapidement dans Amazon Redshift.

Ces avantages sont liés à la manière dont les requêtes d'extraction de données sont AWS SCT créées. Dans certains cas, AWS SCT utilise la fonction analytique DENSE\_RANK dans ces requêtes. DENSE\_RANK peut toutefois utiliser beaucoup de temps et de ressources serveur pour trier le jeu de données résultant de l'extraction. Ainsi, s'il AWS SCT peut fonctionner sans lui, c'est le cas.

Pour trier les données avant de les migrer à l'aide de AWS SCT

1. Ouvrez un AWS SCT projet.
2. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Créer une tâche locale.
3. Choisissez l'onglet Avancé, puis dans Stratégie de tri, choisissez une option :
  - N'utilisez jamais le tri : l'agent d'extraction n'utilise pas la fonction analytique DENSE\_RANK et redémarre depuis le début en cas d'échec.

- Utilisez le tri si possible : l'agent d'extraction utilise DENSE\_RANK si la table possède une clé primaire ou une contrainte unique.
- Utiliser le tri après le premier échec (recommandé) — L'agent d'extraction essaie d'abord d'obtenir les données sans utiliser DENSE\_RANK. Si la première tentative échoue, l'agent d'extraction reconstitue la requête en utilisant DENSE\_RANK et conserve sa position en cas de défaillance.

**Create Local task**

General | **Advanced** | Source server | AWS S3 settings | Source SSL settings

**Extraction settings**

Delimiter character: |

Compression format: GZIP

NULL value as a string

Sorting strategy: Use sorting after first fail (recommen...)

Source temp schema:

Out file size (in MB): 10

Apply RTRIM to string columns

Keep files locally after upload to AWS S3

Use subtasks auto-balancing between agents

Freezing interval: 10

**Copy settings**

Maximum error count: 0

Replace invalid UTF-8 character: ?

Use blank as null value  
BLANKSASNULL: This option loads blank fields, which consist of only white space characters, as NULL. The default behavior, without this option, is to load the space characters as is.

Use empty as null value  
EMPTYASNULL: This option indicates that Amazon Redshift should load empty CHAR and VARCHAR fields as NULL.

Truncate columns  
TRUNCATECOLUMNS: This option truncates data in columns to the appropriate number of characters so that it fits the column specification. This option applies only to columns with a VARCHAR or CHAR data type, and rows 4 MB or less in size.

Automatic compression  
COMPUPDATE: This option controls whether compression encodings are automatically

Test Task | Cancel | Create

4. Définissez des paramètres supplémentaires comme décrit ci-dessous, puis choisissez Create pour créer votre tâche d'extraction de données.

## Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données

Utilisez les procédures suivantes pour créer, exécuter et surveiller des tâches d'extraction de données.

Pour affecter des tâches à des agents et migrer des données

1. Dans le AWS Schema Conversion Tool, après avoir converti votre schéma, choisissez une ou plusieurs tables dans le panneau de gauche de votre projet.

Vous pouvez choisir toutes les tables, cependant, nous ne recommandons pas cette pratique car elle risque d'altérer les performances. Nous vous recommandons de créer plusieurs tâches pour plusieurs tables en fonction de la taille de ces dernières dans votre entrepôt de données.

2. Ouvrez le menu contextuel (clic droit) de chaque table, puis choisissez Créer une tâche. La boîte de dialogue Créer une tâche locale s'ouvre.
3. Dans Nom de la tâche, entrez le nom de la tâche.
4. Pour le mode migration, choisissez l'une des options suivantes :
  - Extraire uniquement : extrayez vos données et enregistrez-les dans vos dossiers de travail locaux.
  - Extraire et chargement : extrayez vos données et chargez-les sur Amazon S3.
  - Extraire, charger et copier : extrayez vos données, chargez-les sur Amazon S3 et copiez-les dans votre entrepôt de données Amazon Redshift.
5. Pour le type de chiffrement, choisissez l'une des options suivantes :
  - AUCUN — Désactivez le chiffrement des données pendant tout le processus de migration des données.
  - CSE\_SK — Utilisez le chiffrement côté client avec une clé symétrique pour migrer les données. AWS SCT génère automatiquement des clés de chiffrement et les transmet aux agents d'extraction de données à l'aide du protocole SSL (Secure Sockets Layer). AWS SCT ne chiffre pas les objets volumineux (LOBs) lors de la migration des données.



6. Choisissez Extraire LOBs pour extraire des objets de grande taille. Si vous n'avez pas besoin d'extraire des objets de grande taille, vous pouvez décocher la case. Cela permet de réduire la quantité de données qui sont extraites.
7. Pour obtenir des informations détaillées sur une tâche, choisissez Activer la journalisation des tâches. Vous pouvez utiliser le journal des tâches pour déboguer les problèmes.

Si vous activez la journalisation des tâches, choisissez le niveau de détails à afficher. Les niveaux sont les suivants, avec chaque niveau comprenant tous les messages du niveau précédent :

- ERROR— Le plus petit détail possible.
  - WARNING
  - INFO
  - DEBUG
  - TRACE— La plus grande quantité de détails.
8. Pour exporter des données depuis BigQuery, AWS SCT utilise le dossier bucket de Google Cloud Storage. Dans ce dossier, les agents d'extraction de données stockent vos données sources.

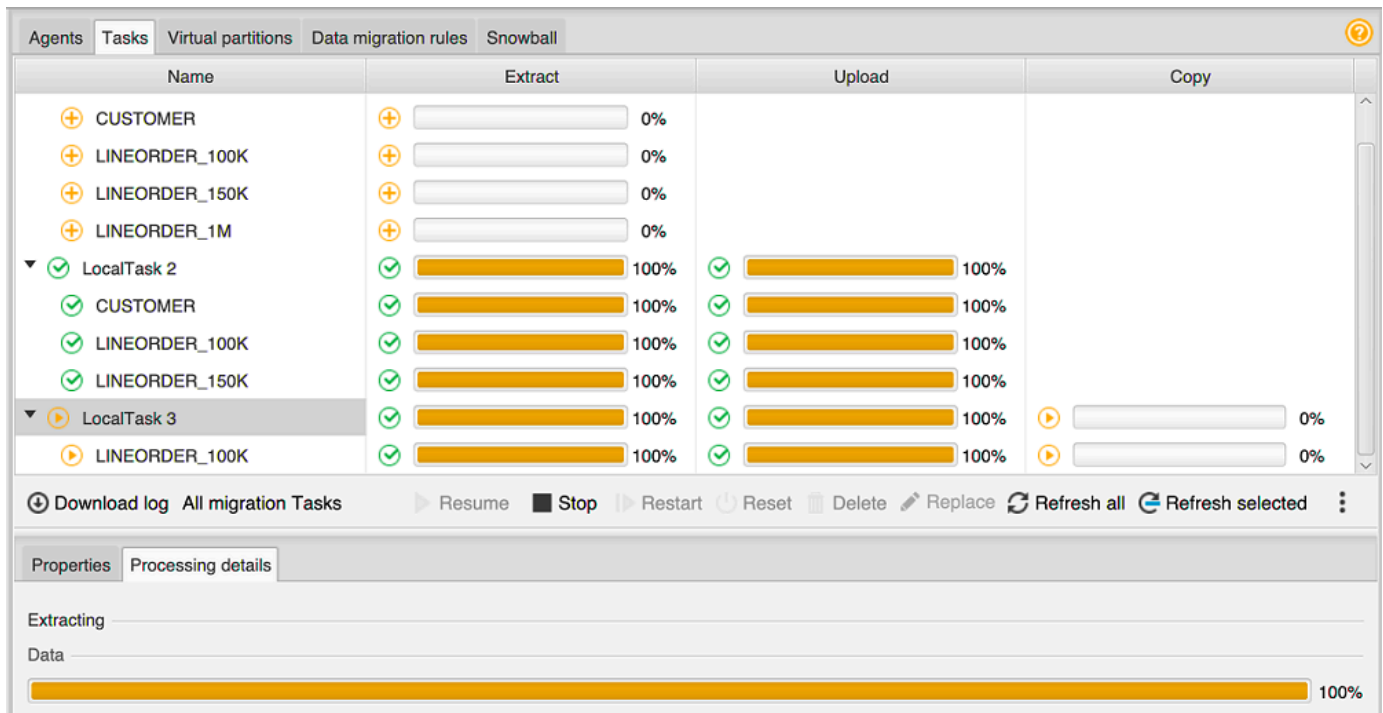
Pour saisir le chemin d'accès à votre dossier de bucket Google Cloud Storage, sélectionnez Avancé. Pour le dossier de bucket Google CS, entrez le nom du bucket et le nom du dossier.

9. Pour jouer le rôle d'utilisateur de votre agent d'extraction de données, choisissez les paramètres Amazon S3. Pour le rôle IAM, entrez le nom du rôle à utiliser. Pour Région, choisissez le Région AWS pour ce rôle.
10. Choisissez Tâche de test pour vérifier que vous pouvez vous connecter à votre dossier de travail, à votre compartiment Amazon S3 et à votre entrepôt de données Amazon Redshift. La vérification dépend du mode de migration que vous avez choisi.
11. Choisissez Create pour créer la tâche.
12. Répétez les étapes précédentes afin de créer des tâches pour toutes les données que vous souhaitez migrer.

## Pour exécuter et surveiller des tâches

1. Pour Afficher, choisissez Vue de migration des données. L'onglet Agents s'affiche.

2. Choisissez l'onglet Tasks. Vos tâches apparaissent dans la grille supérieure, comme illustré ci-après. Vous pouvez voir le statut d'une tâche dans la grille supérieure et celui de ses sous-tâches dans la grille inférieure.



3. Choisissez une tâche dans la grille supérieure et développez-la. Selon le mode de migration que vous avez choisi, vous voyez la tâche divisée en Extract, Upload et Copy.
4. Choisissez Start pour une tâche afin de la démarrer. Vous pouvez surveiller le statut de vos tâches pendant leur exécution. Les sous-tâches s'exécutent en parallèle. Les opérations d'extraction, de chargement et de copie s'exécutent aussi en parallèle.
5. Si vous avez activé la journalisation lorsque vous avez configuré la tâche, vous pouvez afficher le journal:
  - a. Choisissez Télécharger le journal. Un message apparaît avec le nom du dossier qui contient le fichier journal. Ignorez le message.
  - b. Un lien apparaît dans l'onglet Task details. Cliquez sur le lien pour ouvrir le dossier qui contient le fichier journal.

Vous pouvez fermer AWS SCT, et vos agents et tâches continuent de s'exécuter. Vous pouvez rouvrir AWS SCT ultérieurement pour vérifier l'état de vos tâches et consulter les journaux des tâches.

Vous pouvez enregistrer les tâches d'extraction de données sur votre disque local et les restaurer dans le même projet ou dans un autre en utilisant l'exportation et l'importation. Pour exporter une tâche, assurez-vous qu'au moins une tâche d'extraction a été créée dans un projet. Vous pouvez importer une seule tâche d'extraction ou toutes les tâches créées dans le projet.

Lorsque vous exportez une tâche d'extraction, AWS SCT crée un `.xml` fichier distinct pour cette tâche. Le `.xml` fichier stocke les informations de métadonnées de cette tâche, telles que les propriétés, la description et les sous-tâches de la tâche. Le `.xml` fichier ne contient aucune information sur le traitement d'une tâche d'extraction. Les informations suivantes sont recrées lors de l'importation de la tâche :

- Progression de la tâche
- États des sous-tâches et des étapes
- Répartition des agents d'extraction par sous-tâches et étapes
- Tâche et sous-tâche IDs
- Nom de la tâche

## Exportation et importation d'une tâche d'extraction de AWS SCT données

Vous pouvez rapidement enregistrer une tâche existante d'un projet et la restaurer dans un autre projet (ou le même projet) à l'aide de l' AWS SCT exportation et de l'importation. Utilisez la procédure suivante pour exporter et importer des tâches d'extraction de données.

Pour exporter et importer une tâche d'extraction de données

1. Pour Afficher, choisissez Vue de migration des données. L'onglet Agents s'affiche.
2. Choisissez l'onglet Tasks. Vos tâches sont répertoriées dans la grille qui apparaît.
3. Choisissez les trois points alignés verticalement (icône en forme d'ellipse) situés dans le coin inférieur droit sous la liste des tâches.
4. Choisissez Exporter la tâche dans le menu contextuel.
5. Choisissez le dossier dans lequel vous AWS SCT souhaitez placer le `.xml` fichier d'exportation de tâches.

AWS SCT crée le fichier d'exportation de tâches avec le format de nom de fichier `TASK-DESCRIPTION_TASK-ID.xml`.

6. Choisissez les trois points alignés verticalement (icône en forme d'ellipse) en bas à droite sous la liste des tâches.
7. Choisissez Importer une tâche dans le menu contextuel.

Vous pouvez importer une tâche d'extraction dans un projet connecté à la base de données source, et le projet possède au moins un agent d'extraction enregistré actif.

8. Sélectionnez le .xml fichier pour la tâche d'extraction que vous avez exportée.

AWS SCT obtient les paramètres de la tâche d'extraction à partir du fichier, crée la tâche et ajoute la tâche aux agents d'extraction.

9. Répétez ces étapes pour exporter et importer des tâches d'extraction de données supplémentaires.

À la fin de ce processus, votre exportation et votre importation sont terminées et vos tâches d'extraction de données sont prêtes à être utilisées.

## Extraction de données à l'aide d'un appareil AWS Snowball Edge Edge

Le processus d'utilisation AWS SCT d' AWS Snowball Edge Edge comporte plusieurs étapes. La migration implique une tâche locale, qui AWS SCT utilise un agent d'extraction de données pour déplacer les données vers le périphérique AWS Snowball Edge Edge, puis une action intermédiaire qui AWS copie les données de l'appareil AWS Snowball Edge Edge vers un compartiment Amazon S3. Le processus termine le AWS SCT chargement des données depuis le compartiment Amazon S3 vers Amazon Redshift.

Les sections qui suivent cette présentation fournissent un step-by-step guide pour chacune de ces tâches. La procédure suppose que vous avez AWS SCT installé, configuré et enregistré un agent d'extraction de données sur une machine dédiée.

Procédez comme suit pour migrer les données d'un magasin de données local vers un magasin de AWS données à l'aide d' AWS Snowball Edge Edge.

1. Créez une tâche AWS Snowball Edge Edge à l'aide de la AWS Snowball Edge console.
2. Déverrouillez le périphérique AWS Snowball Edge Edge à l'aide de la machine Linux locale dédiée.

3. Créez un nouveau projet dans AWS SCT.
4. Installez et configurez vos agents d'extraction de données.
5. Créez et définissez les autorisations pour le compartiment Amazon S3 à utiliser.
6. Importez une AWS Snowball Edge tâche dans votre AWS SCT projet.
7. Enregistrez votre agent d'extraction de données dans AWS SCT.
8. Créez une tâche locale dans AWS SCT.
9. Exécutez et surveillez la tâche de migration des données dans AWS SCT.

## Step-by-step procédures de migration des données à l'aide AWS SCT d'Edge AWS Snowball Edge

Les sections suivantes fournissent des informations détaillées sur les étapes de la migration.

### Étape 1 : créer une tâche AWS Snowball Edge Edge

Créez un AWS Snowball Edge job en suivant les étapes décrites dans la section [Creating an AWS Snowball Edge Edge Job](#) du Guide du développeur AWS Snowball Edge Edge.

### Étape 2 : déverrouillez l'appareil AWS Snowball Edge Edge

Exécutez les commandes qui déverrouillent et fournissent des informations d'identification à l'appareil Snowball Edge à partir de la machine sur laquelle vous avez installé l' AWS DMS agent. En exécutant ces commandes, vous pouvez être sûr que l'appel de l' AWS DMS agent se connecte au périphérique AWS Snowball Edge Edge. Pour plus d'informations sur le déverrouillage de l'appareil AWS Snowball Edge Edge, voir [Déverrouillage du Snowball Edge](#).

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

### Étape 3 : créer un nouveau AWS SCT projet

Créez ensuite un nouveau AWS SCT projet.

Pour créer un nouveau projet dans AWS SCT

1. Démarrez le AWS Schema Conversion Tool. Dans le menu Fichier, choisissez Nouveau projet. La boîte de dialogue Nouveau projet apparaît.

2. Entrez un nom pour votre projet, qui est stocké localement sur votre ordinateur.
3. Entrez l'emplacement de votre fichier de projet local.
4. Cliquez sur OK pour créer votre AWS SCT projet.
5. Choisissez Ajouter une source pour ajouter une nouvelle base de données source à votre AWS SCT projet.
6. Choisissez Ajouter une cible pour ajouter une nouvelle plateforme cible dans votre AWS SCT projet.
7. Choisissez le schéma de base de données source dans le panneau de gauche.
8. Dans le panneau de droite, spécifiez la plate-forme de base de données cible pour le schéma source sélectionné.
9. Choisissez Créer un mappage. Ce bouton devient actif une fois que vous avez choisi le schéma de base de données source et la plate-forme de base de données cible.

## Étape 4 : Installation et configuration de votre agent d'extraction de données

AWS SCT utilise un agent d'extraction de données pour migrer les données vers Amazon Redshift. Le fichier .zip que vous avez téléchargé pour l'installer AWS SCT inclut le fichier d'installation de l'agent d'extraction. Vous pouvez installer l'agent d'extraction de données sous Windows, Red Hat Enterprise Linux ou Ubuntu. Pour de plus amples informations, veuillez consulter [Installation d'agents d'extraction](#).

Pour configurer votre agent d'extraction de données, entrez vos moteurs de base de données source et cible. Assurez-vous également d'avoir téléchargé les pilotes JDBC pour vos bases de données source et cible sur l'ordinateur sur lequel vous exécutez votre agent d'extraction de données. Les agents d'extraction de données utilisent ces pilotes pour se connecter à vos bases de données source et cible. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

Sous Windows, le programme d'installation de l'agent d'extraction de données lance l'assistant de configuration dans la fenêtre d'invite de commande. Sous Linux, exécutez le `sct-extractor-setup.sh` fichier à partir de l'emplacement où vous avez installé l'agent.

## Étape 5 : Configuration AWS SCT pour accéder au compartiment Amazon S3

Pour plus d'informations sur la configuration d'un compartiment Amazon S3, consultez la [présentation des compartiments](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

## Étape 6 : Importer une AWS Snowball Edge tâche dans votre AWS SCT projet

Pour connecter votre AWS SCT projet à votre appareil AWS Snowball Edge Edge, importez votre AWS Snowball Edge tâche.

Pour importer votre AWS Snowball Edge travail

1. Ouvrez le menu Paramètres, puis choisissez Paramètres globaux. La boîte de dialogue Global settings s'affiche.
2. Choisissez des profils de AWS service, puis choisissez Importer une tâche.
3. Choisissez votre AWS Snowball Edge travail.
4. Entrez votre AWS Snowball Edge adresse IP. Pour plus d'informations, consultez la section [Modification de votre adresse IP](#) dans le guide de AWS Snowball Edge l'utilisateur.
5. Entrez votre AWS Snowball Edge port. Pour plus d'informations, consultez la section [Ports requis pour utiliser les AWS services sur un périphérique AWS Snowball Edge Edge](#) dans le guide du développeur AWS Snowball Edge Edge.
6. Entrez votre cléAWS Snowball Edge d'accès et votre cléAWS Snowball Edge secrète. Pour plus d'informations, voir [Autorisation et contrôle d'accès AWS Snowball Edge dans](#) le guide de AWS Snowball Edge l'utilisateur.
7. Choisissez Appliquer, puis OK.

## Étape 7 : Enregistrez un agent d'extraction de données dans AWS SCT

Dans cette section, vous enregistrez l'agent d'extraction de données dans AWS SCT.

Pour enregistrer un agent d'extraction de données

1. Dans le menu Affichage, choisissez Vue de migration des données (autre), puis sélectionnez Enregistrer.
2. Dans Description, entrez le nom de votre agent d'extraction de données.
3. Dans Nom d'hôte, entrez l'adresse IP de l'ordinateur sur lequel vous exécutez votre agent d'extraction de données.
4. Pour Port, entrez le port d'écoute que vous avez configuré.
5. Choisissez Register (S'inscrire).

## Étape 8 : Création d'une tâche locale

Ensuite, vous créez la tâche de migration. La tâche inclut deux sous-tâches. Une sous-tâche fait migrer les données de la base de données source vers l'apppliance AWS Snowball Edge Edge. L'autre sous-tâche prend les données que l'apppliance charge dans un compartiment Amazon S3 et les migre vers la base de données cible.

Pour créer la tâche de migration

1. Dans le menu Affichage, puis choisissez Affichage de migration des données (autre).
2. Dans le volet gauche où figure le schéma de votre base de données source, choisissez l'objet de schéma à migrer. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Créer une tâche locale.
3. Dans Nom de la tâche, entrez un nom descriptif pour votre tâche de migration de données.
4. Pour le mode migration, choisissez Extraire, uploader et copier.
5. Choisissez les paramètres Amazon S3.
6. Sélectionnez Utiliser Snowball Edge.
7. Entrez des dossiers et des sous-dossiers dans votre compartiment Amazon S3 où l'agent d'extraction de données peut stocker des données.
8. Choisissez Create pour créer la tâche.

## Étape 9 : Exécution et surveillance de la tâche de migration des données dans AWS SCT

Pour démarrer votre tâche de migration de données, sélectionnez Démarrer. Assurez-vous d'avoir établi des connexions à la base de données source, au compartiment Amazon S3, à l' AWS Snowball Edge appareil, ainsi que la connexion à la base de données cible sur AWS.

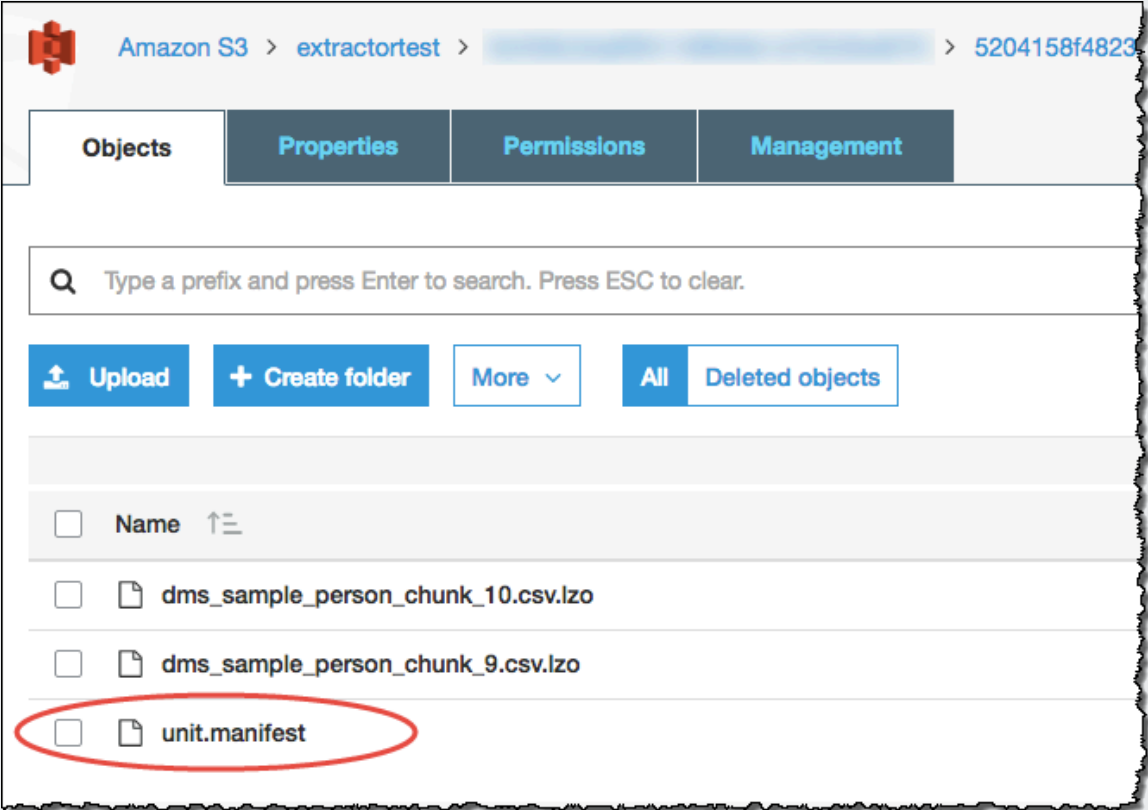
Vous pouvez surveiller et gérer les tâches de migration des données et leurs sous-tâches dans l'onglet Tâches. Vous pouvez suivre la progression de la migration des données, ainsi que suspendre ou redémarrer vos tâches de migration de données.



## Sortie de la tâche d'extraction de données

Une fois vos tâches de migration terminées, vos données sont prêtes. Utilisez les informations suivantes pour déterminer comment continuer en fonction du mode de migration que vous avez choisi et de l'emplacement de vos données.

Mode de migration	Emplacement des données
Extraire, télécharger et copier	<p>Les données se trouvent déjà dans votre entrepôt de données Amazon Redshift. Vous pouvez vérifier que les données sont bien là et commencer à les utiliser. Pour plus d'informations, consultez la section <a href="#">Connexion aux clusters à partir des outils et du code clients</a>.</p>
Extraire et télécharger	<p>Les agents d'extraction ont enregistré vos données sous forme de fichiers dans votre compartiment Amazon S3. Vous pouvez utiliser la commande COPY d'Amazon Redshift pour charger vos données dans Amazon Redshift. Pour plus d'informations, consultez la section <a href="#">Chargement de données depuis Amazon S3</a> dans la documentation Amazon Redshift.</p> <p>Votre compartiment Amazon S3 contient plusieurs dossiers correspondant aux tâches d'extraction que vous avez configurées. Lorsque vous chargez vos données sur Amazon Redshift, spécifiez le nom du fichier manifeste créé par chaque tâche. Le fichier manifeste apparaît dans le dossier des tâches de votre compartiment Amazon S3, comme indiqué ci-dessous.</p>

Mode de migration	Emplacement des données
	 <p>The screenshot shows the Amazon S3 console interface. The breadcrumb path is 'Amazon S3 &gt; extractortest &gt; [redacted] &gt; 5204158f4823'. There are four tabs: 'Objects', 'Properties', 'Permissions', and 'Management'. A search bar contains the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', 'Create folder', 'More', 'All', and 'Deleted objects'. A table of objects is displayed with columns for selection, name, and sorting. The objects listed are 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' object is circled in red.</p>
Extrait uniquement	Les agents d'extraction ont enregistré vos données en tant que fichiers dans votre dossier de travail. Copiez manuellement vos données dans votre compartiment Amazon S3, puis suivez les instructions d'extraction et de chargement.

## Utilisation du partitionnement virtuel avec AWS Schema Conversion Tool

Il est souvent plus facile de gérer les grandes tables non partitionnées en créant des sous-tâches qui créent des partitions virtuelles des données de la table à l'aide des règles de filtrage. Dans AWS SCT, vous pouvez créer des partitions virtuelles pour vos données migrées. Il existe trois types de partitions, qui fonctionnent avec des types de données spécifiques :

- Le type de partition RANGE fonctionne avec les types de données numérique, de date et d'heure.

- Le type de partition LIST fonctionne avec les types de données numérique, de caractère, de date et d'heure.
- Le type de partition DATE AUTO SPLIT fonctionne avec les types de données numériques, de date et d'heure.

AWS SCT valide les valeurs que vous fournissez pour créer une partition. Par exemple, si vous tentez de partitionner une colonne avec le type de données NUMERIC mais que vous fournissez des valeurs d'un autre type de données, une AWS SCT erreur est générée.

En outre, si vous migrez des données AWS SCT vers Amazon Redshift, vous pouvez utiliser le partitionnement natif pour gérer la migration de tables volumineuses. Pour de plus amples informations, veuillez consulter [Utilisation du partitionnement natif](#).

## Limites lors de la création d'un partitionnement virtuel

Les limitations à la création d'une partition virtuelle sont les suivantes :

- Vous ne pouvez utiliser le partitionnement virtuel que pour les tables non partitionnées.
- Vous ne pouvez utiliser le partitionnement virtuel que dans la vue de migration des données.
- Vous ne pouvez pas utiliser l'option UNION ALL VIEW avec le partitionnement virtuel.

## Type de cloison RANGE

Le type de partition RANGE partitionne les données en fonction d'une plage de valeurs de colonne pour les types de données numérique, de date et d'heure. Ce type de partition crée une clause WHERE et vous fournissez la plage de valeurs pour chaque partition. Pour spécifier une liste de valeurs pour la colonne partitionnée, utilisez la zone Valeurs. Vous pouvez charger les informations de valeur en utilisant un fichier .csv.

Le type de partition RANGE crée des partitions par défaut aux deux extrémités des valeurs de partition. Ces partitions par défaut capturent toutes les données inférieures ou supérieures aux valeurs de partition spécifiées.

Par exemple, vous pouvez créer plusieurs partitions en fonction d'une plage de valeurs que vous fournissez. Dans l'exemple suivant, les valeurs de partitionnement de LO\_TAX sont spécifiées pour créer plusieurs partitions.

```
Partition1: WHERE LO_TAX <= 10000.9
```

```
Partition2: WHERE LO_TAX > 10000.9 AND LO_TAX <= 15005.5
```

```
Partition3: WHERE LO_TAX > 15005.5 AND LO_TAX <= 25005.95
```

## Pour créer une partition virtuelle RANGE

1. Ouvrez AWS SCT.
2. Choisissez le mode d'affichage de migration des données (autre).
3. Choisissez la table dans laquelle vous souhaitez configurer le partitionnement virtuel. Ouvrez le menu contextuel (clic droit) de la table, puis choisissez Ajouter un partitionnement virtuel.
4. Dans la boîte de dialogue Ajouter un partitionnement virtuel, entrez les informations suivantes.

Option	Action
Type de cloison	Choisissez RANGE. L'interface utilisateur de la boîte de dialogue varie en fonction du type que vous choisissez.
Nom de colonne	Choisissez la colonne à partitionner.
Type de colonne	Choisissez le type de données des valeurs de la colonne.
Valeurs	Ajoutez de nouvelles valeurs en tapant chaque valeur dans la zone New Value, puis en choisissant le signe plus pour ajouter la valeur.
Charger depuis un fichier	(Facultatif) Indiquez le nom du fichier .csv contenant les valeurs de partition.

5. Choisissez OK.

## Type de partition LIST

Le type de partition LIST partitionne les données en fonction de valeurs de colonne pour les types de données numérique, de caractère, de date et d'heure. Ce type de partition crée une clause WHERE et vous fournissez les valeurs pour chaque partition. Pour spécifier une liste de valeurs pour la colonne partitionnée, utilisez la zone Valeurs. Vous pouvez charger les informations de valeur en utilisant un fichier .csv.

Par exemple, vous pouvez créer plusieurs partitions en fonction d'une valeur que vous fournissez. Dans l'exemple suivant, les valeurs de partitionnement de LO\_ORDERKEY sont spécifiées pour créer plusieurs partitions.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Vous pouvez également créer une partition par défaut pour les valeurs non incluses dans celles spécifiées.

Vous pouvez utiliser le type de partition LIST pour filtrer les données sources si vous souhaitez exclure des valeurs spécifiques de la migration. Supposons, par exemple, que vous souhaitiez omettre les lignes avec LO\_ORDERKEY = 4. Dans ce cas, n'incluez pas la valeur 4 dans la liste des valeurs de partition et assurez-vous que l'option Inclure d'autres valeurs n'est pas sélectionnée.

Pour créer une partition virtuelle LIST

1. Ouvrez AWS SCT.
2. Choisissez le mode d'affichage de migration des données (autre).
3. Choisissez la table dans laquelle vous souhaitez configurer le partitionnement virtuel. Ouvrez le menu contextuel (clic droit) de la table, puis choisissez Ajouter un partitionnement virtuel.
4. Dans la boîte de dialogue Ajouter un partitionnement virtuel, entrez les informations suivantes.

Option	Action
Type de cloison	Choisissez LIST. L'interface utilisateur de la boîte de dialogue varie en fonction du type que vous choisissez.
Nom de colonne	Choisissez la colonne à partitionner.
Nouvelle valeur	Tapez ici une valeur à ajouter à l'ensemble de valeurs de partitionnement.

Option	Action
Inclure d'autres valeurs	Choisissez cette option pour créer une partition par défaut dans laquelle sont stockées toutes les valeurs qui ne répondent aux critères de partitionnement.
Charger depuis un fichier	(Facultatif) Indiquez le nom du fichier .csv contenant les valeurs de partition.

5. Choisissez OK.

## Type de partition DATE AUTO SPLIT

Le type de partition DATE AUTO SPLIT est un moyen automatique de générer des partitions RANGE. Avec DATA AUTO SPLIT, vous indiquez AWS SCT l'attribut de partitionnement, les points de départ et de fin, ainsi que la taille de la plage entre les valeurs. AWS SCT Calcule ensuite automatiquement les valeurs de partition.

DATA AUTO SPLIT automatise une grande partie du travail lié à la création de partitions de plage. Le compromis entre l'utilisation de cette technique et le partitionnement par plage est le niveau de contrôle dont vous avez besoin sur les limites de la partition. Le processus de division automatique crée toujours des plages de tailles égales (uniformes). Le partitionnement des plages vous permet de faire varier la taille de chaque plage en fonction de votre distribution de données particulière. Par exemple, vous pouvez utiliser tous les jours, toutes les semaines, toutes les deux semaines, tous les mois, etc.

```
Partition1: WHERE LO_ORDERDATE >= '1954-10-10' AND LO_ORDERDATE < '1954-10-24'
Partition2: WHERE LO_ORDERDATE >= '1954-10-24' AND LO_ORDERDATE < '1954-11-06'
Partition3: WHERE LO_ORDERDATE >= '1954-11-06' AND LO_ORDERDATE < '1954-11-20'
...
PartitionN: WHERE LO_ORDERDATE >= USER_VALUE_N AND LO_ORDERDATE <= '2017-08-13'
```

### Pour créer une partition virtuelle DATE AUTO SPLIT

1. Ouvrez AWS SCT.
2. Choisissez le mode d'affichage de migration des données (autre).
3. Choisissez la table dans laquelle vous souhaitez configurer le partitionnement virtuel. Ouvrez le menu contextuel (clic droit) de la table, puis choisissez Ajouter un partitionnement virtuel.

4. Dans la boîte de dialogue Ajouter un partitionnement virtuel, entrez les informations suivantes.

Option	Action
Type de cloison	Choisissez DATE AUTO SPLIT. L'interface utilisateur de la boîte de dialogue varie en fonction du type que vous choisissez.
Nom de colonne	Choisissez la colonne à partitionner.
Date de début	Tapez une date de début.
Date de fin	Tapez une date de fin.
Intervalle	Entrez l'unité d'intervalle et choisissez la valeur de cette unité.

5. Choisissez OK.

## Utilisation du partitionnement natif

Pour accélérer la migration des données, vos agents d'extraction de données peuvent utiliser des partitions natives de tables sur votre serveur d'entrepôt de données source. AWS SCT prend en charge le partitionnement natif pour les migrations de Greenplum, Netezza et Oracle vers Amazon Redshift.

Par exemple, après avoir créé un projet, vous pouvez collecter des statistiques sur un schéma et analyser la taille des tables sélectionnées pour la migration. Pour les tables dont la taille dépasse la taille spécifiée, AWS SCT déclenche le mécanisme de partitionnement natif.

Pour utiliser le partitionnement natif

1. Ouvrez AWS SCT et choisissez Nouveau projet pour Fichier. La boîte de dialogue Nouveau projet apparaît.
2. Créez un nouveau projet, ajoutez vos serveurs source et cible et créez des règles de mappage. Pour de plus amples informations, veuillez consulter [Démarrage et gestion de projets dans AWS SCT](#).
3. Choisissez Affichage, puis sélectionnez Affichage principal.
4. Pour les paramètres du projet, choisissez l'onglet Migration des données. Choisissez Utiliser le partitionnement automatique. Pour les bases de données sources Greenplum et Netezza, entrez

la taille minimale des tables prises en charge en mégaoctets (par exemple, 100). AWS SCT crée automatiquement des sous-tâches de migration distinctes pour chaque partition native qui n'est pas vide. Pour les migrations d'Oracle vers Amazon Redshift, AWS SCT crée des sous-tâches pour toutes les tables partitionnées.

5. Dans le panneau de gauche qui affiche le schéma de votre base de données source, choisissez-en un. Ouvrez le menu contextuel (clic droit) de l'objet, puis choisissez Collecter des statistiques. Pour la migration des données d'Oracle vers Amazon Redshift, vous pouvez ignorer cette étape.
6. Choisissez toutes les tables à migrer.
7. Enregistrez le nombre d'agents requis. Pour de plus amples informations, veuillez consulter [Enregistrement des agents d'extraction auprès du AWS Schema Conversion Tool](#).
8. Créez une tâche d'extraction de données pour les tables sélectionnées. Pour de plus amples informations, veuillez consulter [Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données](#).

Vérifiez si les grandes tables sont divisées en sous-tâches et que chaque sous-tâche correspond au jeu de données qui présente une partie de la table située sur une tranche de votre entrepôt de données source.

9. Démarrez et surveillez le processus de migration jusqu'à ce que AWS SCT les agents d'extraction de données aient terminé la migration des données depuis vos tables sources.

## Migration LOBs vers Amazon Redshift

Amazon Redshift ne prend pas en charge le stockage d'objets binaires volumineux (LOBs). Toutefois, si vous devez en migrer une ou plusieurs LOBs vers Amazon Redshift, AWS SCT vous pouvez effectuer la migration. Pour ce faire, AWS SCT utilise un compartiment Amazon S3 pour stocker LOBs et écrit l'URL du compartiment Amazon S3 dans les données migrées stockées dans Amazon Redshift.

Pour migrer LOBs vers Amazon Redshift

1. Ouvrez un AWS SCT projet.
2. Connectez-vous aux bases de données source et cible. Actualisez les métadonnées de la base de données cible et assurez-vous que les tables converties y existent.
3. Pour Actions, choisissez Créer une tâche locale.
4. Pour le mode migration, choisissez l'une des options suivantes :



- Extrayez et chargez pour extraire vos données, et chargez vos données sur Amazon S3.
  - Extrayez, chargez et copiez pour extraire vos données, chargez vos données sur Amazon S3 et copiez-les dans votre entrepôt de données Amazon Redshift.
5. Choisissez les paramètres Amazon S3.
  6. Pour LOBs le dossier du compartiment Amazon S3, entrez le nom du dossier dans le compartiment Amazon S3 où vous souhaitez le LOBs stocker.

Si vous utilisez le profil de AWS service, ce champ est facultatif. AWS SCT peut utiliser les paramètres par défaut de votre profil. Pour utiliser un autre compartiment Amazon S3, entrez le chemin ici.

7. Activez l'option Utiliser un proxy pour utiliser un serveur proxy afin de télécharger des données sur Amazon S3. Choisissez ensuite le protocole de transfert de données, entrez le nom d'hôte, le port, le nom d'utilisateur et le mot de passe.
8. Pour le type de point de terminaison, choisissez FIPS pour utiliser le point de terminaison FIPS (Federal Information Processing Standard). Choisissez VPCE pour utiliser le point de terminaison du cloud privé virtuel (VPC). Ensuite, pour le point de terminaison VPC, entrez le système de nom de domaine (DNS) de votre point de terminaison VPC.
9. Activez l'option Conserver les fichiers sur Amazon S3 après les avoir copiés sur Amazon Redshift pour conserver les fichiers extraits sur Amazon S3 après les avoir copiés sur Amazon Redshift.
10. Choisissez Create pour créer la tâche.

## Bonnes pratiques et résolution des problèmes pour les agents d'extraction de données

Voici quelques bonnes pratiques et suggestions de dépannage relatives à l'utilisation des agents d'extraction.

Problème	Suggestions de dépannage
Les performances sont lentes	Pour améliorer les performances, nous vous recommandons de procéder comme suit : <ul style="list-style-type: none"><li>• Installez plusieurs agents.</li></ul>

Problème	Suggestions de dépannage
	<ul style="list-style-type: none"><li>• Installez des agents sur des ordinateurs proches de votre entrepôt de données.</li><li>• N'exécutez pas toutes les tables sur une seule tâche d'agent.</li></ul>
Délais d'attente en cas de conflit	Évitez d'avoir un nombre trop élevé d'agents accédant simultanément à votre entrepôt de données.
Un agent s'arrête temporairement	Si un agent s'arrête, chacune de ses tâches apparaît comme ayant échoué dans AWS SCT. Si vous attendez un peu, il arrive parfois qu'une reprise de l'agent se produise. Dans ce cas, le statut de ses tâches est mis à jour dans AWS SCT.
Un agent s'arrête définitivement	<p>Si l'ordinateur qui exécute un agent s'arrête de façon définitive alors que celui-ci est en train d'exécuter une tâche, vous pouvez remplacer cet agent par un autre afin de poursuivre l'exécution de la tâche. Vous pouvez le remplacer par un nouvel agent uniquement si le dossier de travail de l'agent d'origine ne se trouve pas sur le même ordinateur que l'agent d'origine. Pour le remplacer par un nouvel agent, procédez comme suit :</p> <ul style="list-style-type: none"><li>• Installez un agent sur un nouvel ordinateur.</li><li>• Configurez le nouvel agent avec les mêmes paramètres, y compris le numéro de port et le dossier de travail, que l'agent d'origine.</li><li>• Démarrez l'agent. Une fois l'agent démarré, la tâche découvre le nouvel agent disponible et continue de s'exécuter sur celui-ci.</li></ul>

# Conversion du code SQL d'une application en utilisant AWS SCT

Lorsque vous convertissez votre schéma de base de données à partir d'un moteur à un autre, vous devez également mettre à jour le code SQL dans vos applications pour interagir avec le nouveau moteur de base de données au lieu de l'ancien. Vous pouvez afficher, analyser, modifier et enregistrer le code SQL converti.

Vous pouvez utiliser le AWS Schema Conversion Tool (AWS SCT) pour convertir le code SQL dans votre code d'application C++, C#, Java ou autre. Pour une conversion d'Oracle vers PostgreSQL, vous pouvez l' AWS SCT utiliser pour convertir le code SQL\*Plus en PSQL. En outre, pour une conversion d'Oracle vers PostgreSQL, vous pouvez l' AWS SCT utiliser pour convertir le code SQL intégré dans les applications C#, C++, Java et Pro\*C.

## Rubriques

- [Présentation de la conversion du code SQL d'une application](#)
- [Convertissez le code SQL dans vos applications avec AWS SCT](#)
- [Conversion de code SQL dans des applications C# avec AWS Schema Conversion Tool](#)
- [Conversion de code SQL dans des applications C++ avec AWS Schema Conversion Tool](#)
- [Conversion de code SQL dans des applications Java avec AWS Schema Conversion Tool](#)
- [Conversion de code SQL dans les applications Pro\\*C avec AWS Schema Conversion Tool](#)

## Présentation de la conversion du code SQL d'une application

Pour convertir le code SQL dans votre application, suivez les étapes de haut niveau suivantes :

- Créer un projet de conversion d'application : le projet de conversion d'application est un enfant du projet de conversion de schéma de base de données. Chaque projet de conversion de schéma de base de données peut avoir un ou plusieurs projets de conversion d'application enfant. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications génériques dans AWS SCT](#).
- Analysez et convertissez votre code SQL : AWS SCT analyse votre application, extrait le code SQL et crée une version locale du code SQL converti que vous pouvez consulter et modifier. L'outil ne change pas le code de votre application jusqu'à ce que vous soyez prêt. Pour de plus amples informations, veuillez consulter [Analyse et conversion de votre code SQL dans AWS SCT](#).

- Création d'un rapport d'évaluation des applications : le rapport d'évaluation des applications fournit des informations importantes sur la conversion du code SQL de l'application de votre schéma de base de données source vers le schéma de base de données cible. Pour de plus amples informations, veuillez consulter [Création et utilisation du rapport AWS SCT d'évaluation dans AWS SCT](#).
- Modifiez, appliquez des modifications et enregistrez votre code SQL converti : le rapport d'évaluation inclut une liste d'éléments de code SQL qui ne peuvent pas être convertis automatiquement. Pour ces éléments, vous pouvez modifier le code SQL manuellement pour exécuter la conversion. Pour de plus amples informations, veuillez consulter [Modifier et enregistrer votre code SQL converti avec AWS SCT](#).

## Convertissez le code SQL dans vos applications avec AWS SCT

Vous pouvez l'utiliser AWS SCT pour convertir le code SQL intégré à vos applications. Le convertisseur AWS SCT d'applications générique traite le code de votre application comme du texte brut. Il analyse le code de votre application et extrait le code SQL à l'aide d'expressions régulières. Ce convertisseur prend en charge différents types de fichiers de code source et fonctionne avec le code d'application écrit dans n'importe quel langage de programmation.

Le convertisseur d'applications génériques présente les limites suivantes. Il n'explore pas en profondeur la logique de l'application spécifique au langage de programmation de votre application. De plus, le convertisseur générique ne prend pas en charge les instructions SQL provenant de différents objets d'application, tels que les fonctions, les paramètres, les variables locales, etc.

Pour améliorer la conversion du code SQL de votre application, utilisez des convertisseurs de code SQL d'application spécifiques au langage. Pour plus d'informations, consultez [Code SQL dans les applications C#](#), [Code SQL en Java](#) et [Code SQL dans Pro\\*C](#).

## Création de projets de conversion d'applications génériques dans AWS SCT

Dans le AWS Schema Conversion Tool, le projet de conversion de l'application est un enfant du projet de conversion du schéma de base de données. Chaque projet de conversion de schéma de base de données peut avoir un ou plusieurs projets de conversion d'application enfant.

### Note

AWS SCT ne prend pas en charge la conversion entre les sources et cibles suivantes :

- Oracle vers Oracle
- PostgreSQL vers PostgreSQL ou Aurora PostgreSQL
- MySQL vers MySQL
- SQL Server vers SQL Server
- Amazon Redshift vers Amazon Redshift
- SQL Server vers Babelfish
- Services d'intégration de SQL Server pour AWS Glue
- Apache Cassandra vers Amazon DynamoDB

Utilisez la procédure suivante pour créer un projet de conversion d'application générique.

Pour créer un projet de conversion d'application

1. Dans le menu Applications AWS Schema Conversion Tool, choisissez Nouvelle application générique.

La boîte de dialogue New application conversion project s'affiche.

## 2. Ajoutez les informations du projet suivantes.

Pour ce paramètre	Faites ceci
Nom	Entrez un nom pour votre projet de conversion d'applications. Chaque projet de conversion du schéma de base de données peut avoir un ou plusieurs projets de conversion d'application enfant. Choisissez par conséquent un nom logique si vous ajoutez plus de projets plus tard.
Emplacement	Saisissez l'emplacement du code source pour votre application.
Langue	Sélectionnez l'une des méthodes suivantes : <ul style="list-style-type: none"> <li>• Java</li> </ul>

Pour ce paramètre	Faites ceci
	<ul style="list-style-type: none"> <li>• C++</li> <li>• C#</li> <li>• Any</li> </ul>
Style du paramètre cible	<p>Choisissez la syntaxe à utiliser pour les variables de liaison dans le code converti. Les différentes plateformes de base de données utilisent une syntaxe différente pour les variables de liaison. Choisissez l'une des options suivantes :</p> <ul style="list-style-type: none"> <li>• Same as in source</li> <li>• Positional (?)</li> <li>• Indexed (:1)</li> <li>• Indexed (\$1)</li> <li>• Named (@name)</li> <li>• Named (:name)</li> <li>• Nommé (&amp;name)</li> <li>• Nommé (\$name)</li> <li>• Nommé (#name)</li> <li>• Nommé (! nom !)</li> </ul>
Choisissez le schéma de base de données source	Dans l'arborescence des sources, choisissez le schéma utilisé par votre application. Assurez-vous que ce schéma fait partie d'une règle de mappage.

3. Sélectionnez Ne pas convertir les variables de liaison en types SQL pour éviter la conversion des types de variables de liaison en types SQL. Cette option n'est disponible que pour une conversion d'Oracle vers PostgreSQL.

Par exemple, le code source de votre application inclut la requête Oracle suivante :

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Lorsque vous sélectionnez Ne pas convertir les variables de liaison en types SQL, AWS SCT convertit cette requête comme indiqué ci-dessous.

```
SELECT * FROM account WHERE id = ?
```

Lorsque vous désactivez la case Ne pas convertir les variables de liaison en types SQL, AWS SCT le type de variable de liaison est remplacé par le type de NUMERIC données. Le résultat de la conversion est affiché ci-dessous.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

4. Sélectionnez Conserver les noms d'objets pour éviter d'ajouter le nom du schéma au nom de l'objet converti. Cette option n'est disponible que pour une conversion d'Oracle vers PostgreSQL.

Supposons, par exemple, que le code source de votre application inclut la requête Oracle suivante.

```
SELECT * FROM ACCOUNT
```

Lorsque vous sélectionnez Conserver les noms d'objets, AWS SCT convertit cette requête comme indiqué ci-dessous.

```
SELECT * FROM account
```

Lorsque vous désactivez l'option Conserver les noms d'objets, AWS SCT ajoute le nom du schéma au nom de la table. Le résultat de la conversion est affiché ci-dessous.

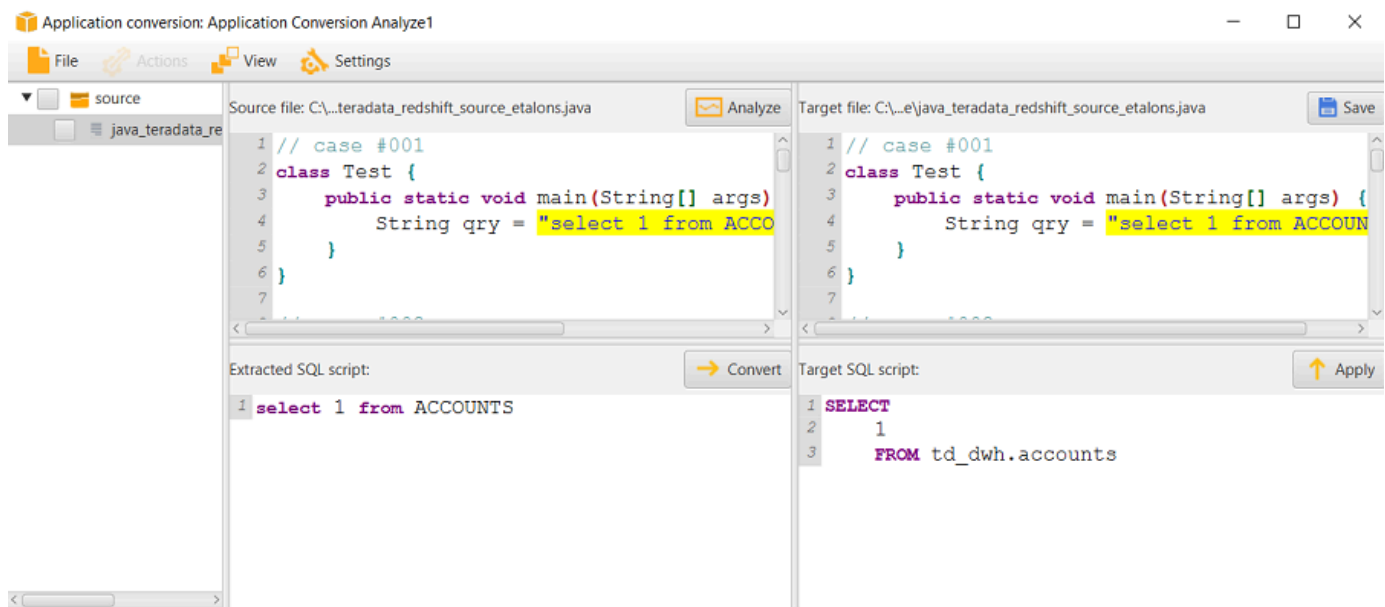
```
SELECT * FROM schema_name.account
```

Si votre code source inclut les noms des objets parents dans les noms des objets, AWS SCT utilise ce format dans le code converti. Dans ce cas, ignorez l'option Conserver les noms des objets car elle AWS SCT ajoute les noms des objets parents dans le code converti.

5. Choisissez OK pour créer votre projet de conversion d'application.

La fenêtre de projet s'ouvre.





## Gestion des projets de conversion d'applications dans AWS SCT

Vous pouvez ouvrir un projet de conversion d'applications existant et ajouter plusieurs projets de conversion d'applications.

Après avoir créé un projet de conversion d'applications, la fenêtre du projet s'ouvre automatiquement. Vous pouvez fermer la fenêtre du projet de conversion de l'application et y revenir ultérieurement.

Pour ouvrir un projet de conversion d'applications existant

1. Dans le panneau de gauche, choisissez le nœud du projet de conversion d'applications et ouvrez le menu contextuel (clic droit).
2. Choisissez Gérer l'application.

Pour ajouter un projet de conversion d'applications supplémentaire

1. Dans le panneau de gauche, choisissez le nœud du projet de conversion d'applications et ouvrez le menu contextuel (clic droit).
2. Choisissez New application (Nouvelle application).
3. Entrez les informations requises pour créer un nouveau projet de conversion d'applications. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications génériques](#).

## Analyse et conversion de votre code SQL dans AWS SCT

Utilisez la procédure suivante pour analyser et convertir votre code SQL dans le AWS Schema Conversion Tool.

Pour analyser et convertir votre code SQL

1. Ouvrez un projet de conversion d'applications existant, puis choisissez Analyser.

AWS SCT analyse le code de votre application et extrait le code SQL. AWS SCT affiche le code SQL extrait dans la liste des scripts SQL analysés.

2. Pour les scripts SQL analysés, choisissez un élément pour consulter le code SQL extrait. AWS SCT affiche le code de l'élément sélectionné dans le volet du script SQL extrait.
3. Choisissez Convertir pour convertir le code SQL dans le volet de script SQL extrait. AWS SCT convertit le code dans un format compatible avec votre base de données cible.

Vous pouvez modifier le code SQL converti. Pour de plus amples informations, veuillez consulter [Modification et enregistrement de votre code SQL converti](#).

Source file	Position	Extracted code	Converted code
C:\And...s.java	Line 11 22:50	✓ "select 1 a from ACCOUNTS b"	✓ SELECT 1 AS a FROM td_dwh.accounts AS b
C:\And...s.java	Line 18 22:46	✓ "select * from ACCOUNTS"	
C:\And...s.java	Line 25 22:50	✓ "select a.* from ACCOUNTS a"	

4. Lorsque vous créez un rapport d'évaluation de la conversion d'une application, il AWS SCT convertit tous les éléments de code SQL extraits. Pour de plus amples informations, veuillez consulter [Création et utilisation du rapport d'évaluation](#).

# Création et utilisation du rapport AWS SCT d'évaluation dans AWS SCT

Le rapport d'évaluation de la conversion des applications fournit des informations sur la conversion du code SQL de l'application dans un format compatible avec votre base de données cible. Le rapport détaille tout le code SQL extrait, tout le code SQL converti et les actions à effectuer pour le code SQL qui ne AWS SCT peut pas être converti.

## Création d'un rapport d'évaluation de la conversion des applications

Utilisez la procédure suivante pour créer un rapport d'évaluation de la conversion des applications.

Pour créer un rapport d'évaluation de la conversion des applications

1. Dans la fenêtre du projet de conversion de l'application, choisissez Créer un rapport dans le menu Actions.

AWS SCT crée le rapport d'évaluation de la conversion des applications et l'ouvre dans la fenêtre du projet de conversion des applications.

2. Vérifiez l'onglet Summary.

L'onglet Résumé, illustré ci-dessous, affiche les informations récapitulatives du rapport d'évaluation de l'application. Il indique les éléments du code SQL qui ont été convertis automatiquement et ceux qui ne l'ont pas été.



3. Choisissez les actions d'extraction SQL.

Consultez la liste des éléments de code SQL qui ne AWS SCT peuvent pas être extraits de votre code source.

4. Choisissez les actions de conversion SQL.

Consultez la liste des éléments de code SQL qui ne AWS SCT peuvent pas être convertis automatiquement. Utilisez les actions recommandées pour convertir manuellement le code SQL. Pour plus d'informations sur la façon de modifier votre code SQL converti, consultez [Modifier et enregistrer votre code SQL converti avec AWS SCT](#).

5. (Facultatif) Enregistrez une copie locale du rapport sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :

- Choisissez Enregistrer au format PDF en haut à droite pour enregistrer le rapport sous forme de fichier PDF.

Le fichier PDF contient le résumé, les mesures à prendre et les recommandations relatives à la conversion des applications.

- Choisissez Enregistrer au format CSV en haut à droite pour enregistrer le rapport sous forme de fichier CSV.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir le code SQL.

## Modifier et enregistrer votre code SQL converti avec AWS SCT

Le rapport d'évaluation inclut une liste des éléments de code SQL qui ne AWS SCT peuvent pas être convertis. Pour chaque élément, AWS SCT crée une action dans l'onglet Actions de conversion SQL. Pour ces éléments, vous pouvez modifier le code SQL manuellement pour exécuter la conversion.

Utilisez la procédure suivante pour modifier votre code SQL converti, appliquer les modifications et les enregistrer.

Pour modifier, appliquer des modifications à votre code converti de SQL et l'enregistrer

1. Modifiez votre code SQL converti directement dans le volet Target SQL script. S'il n'y a aucun code converti affiché, vous pouvez cliquer dans le volet et commencer à taper.
2. Une fois que vous avez terminé de modifier votre code SQL converti, choisissez Apply. A ce stade, les modifications sont enregistrées dans la mémoire, mais pas encore écrites dans votre fichier.
3. Choisissez Save pour enregistrer vos modifications dans votre fichier.

Lorsque vous choisissez Enregistrer, vous remplacez votre fichier d'origine. Faites une copie de votre fichier d'origine avant d'enregistrer afin de conserver un enregistrement de votre code d'application d'origine.

# Conversion de code SQL dans des applications C# avec AWS Schema Conversion Tool

Pour une conversion d'Oracle vers PostgreSQL, vous pouvez utiliser AWS Schema Conversion Tool pour convertir le code SQL intégré dans vos applications C#. Ce convertisseur d'application C# spécifique comprend la logique de l'application. Il collecte des instructions situées dans différents objets d'application, tels que des fonctions, des paramètres, des variables locales, etc.

Grâce à cette analyse approfondie, le convertisseur de code SQL de l'application C# fournit de meilleurs résultats de conversion que le convertisseur générique.

## Création de projets de conversion d'applications C# dans AWS SCT

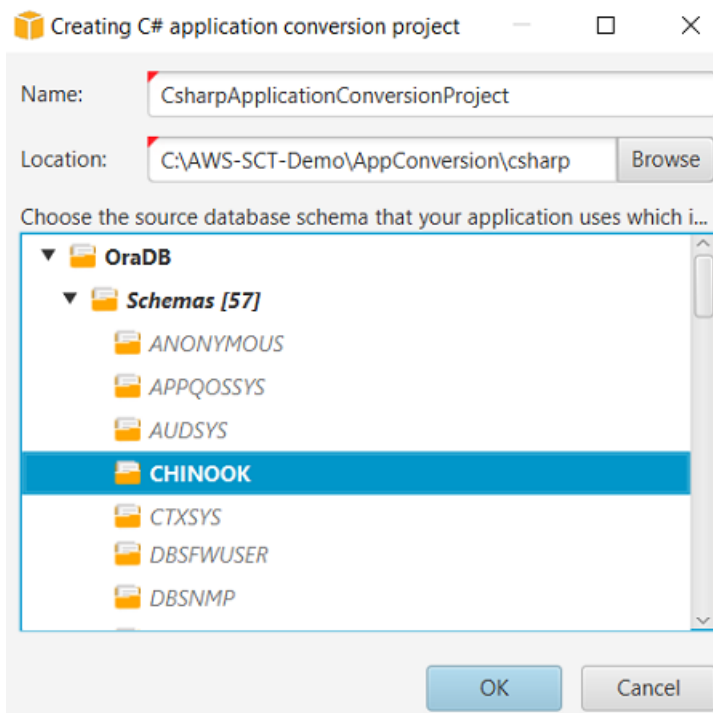
Vous pouvez créer un projet de conversion d'application C# uniquement pour convertir des schémas de base de données Oracle en schémas de base de données PostgreSQL. Assurez-vous d'ajouter une règle de mappage dans votre projet qui inclut un schéma Oracle source et une base de données PostgreSQL cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Vous pouvez ajouter plusieurs projets de conversion d'applications dans un seul AWS SCT projet. Utilisez la procédure suivante pour créer un projet de conversion d'application C#.

Pour créer un projet de conversion d'applications C#

1. Créez un projet de conversion de base de données et ajoutez une base de données Oracle source. Pour plus d'informations, consultez [Démarrage et gestion de projets dans AWS SCT](#) et [Ajouter des serveurs au projet dans AWS SCT](#).
2. Ajoutez une règle de mappage qui inclut votre base de données Oracle source et une base de données PostgreSQL cible. Vous pouvez ajouter une base de données PostgreSQL cible ou utiliser une plate-forme de base de données cible PostgreSQL virtuelle dans une règle de mappage. Pour plus d'informations, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#) et [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).
3. Dans le menu Affichage, choisissez Vue principale.
4. Dans le menu Applications, choisissez Nouvelle application C#.

La boîte de dialogue Création d'un projet de conversion d'application C# apparaît.



5. Dans Nom, entrez le nom de votre projet de conversion d'application C#. Chaque projet de conversion de schéma de base de données pouvant comporter un ou plusieurs projets de conversion d'applications secondaires, choisissez un nom pertinent si vous ajoutez plusieurs projets.
6. Dans Emplacement, entrez l'emplacement du code source de votre application.
7. Dans l'arborescence des sources, choisissez le schéma utilisé par votre application. Assurez-vous que ce schéma fait partie d'une règle de mappage. AWS SCT met en évidence les schémas qui font partie d'une règle de mappage en gras.
8. Cliquez sur OK pour créer votre projet de conversion d'application C#.
9. Trouvez votre projet de conversion d'applications C# dans le nœud Applications du panneau de gauche.

## Conversion du code SQL de votre application C# en AWS SCT

Après avoir ajouté votre application C# au AWS SCT projet, convertissez le code SQL de cette application dans un format compatible avec votre plate-forme de base de données cible. Utilisez la procédure suivante pour analyser et convertir le code SQL intégré à votre application C# dans le AWS Schema Conversion Tool.

## Pour convertir votre code SQL

1. Développez le nœud C# sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir. AWS SCT analyse vos fichiers de code source, détermine la logique de l'application et charge les métadonnées du code dans le projet. Ces métadonnées de code incluent les classes C#, les objets, les méthodes, les variables globales, les interfaces, etc.

Dans le panneau de base de données cible, AWS SCT crée une structure de dossiers similaire à celle de votre projet d'application source. Vous pouvez consulter ici le code d'application converti.

```
Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18      {
19          string str1 = "SELECT *\n" +
20                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
21                      "WHERE COL_CHAR = \'CHAR\'';
22
23          command.CommandText = str1;
24          command.ExecuteNonQuery();
25      }
26      connection.Close();
Cursor position: 1005

Target Amazon RDS for PostgreSQL file: SpecialEscapeSequences.cs
Properties Text Apply status Key management
19      {
20          string str1 = "SELECT *\n" +
21                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
22                      "WHERE COL_CHAR = \'CHAR\'';
23
24          command.CommandText = str1;
25          command.ExecuteNonQuery();
26      }
27      connection.Close();
```

4. Enregistrez le code d'application converti. Pour de plus amples informations, veuillez consulter [Sauvegarde du code d'application converti](#).

Vos applications C# peuvent inclure du code SQL qui interagit avec différentes bases de données sources. Vous pouvez migrer vers PostgreSQL plusieurs de ces bases de données sources. Dans ce cas, assurez-vous de ne pas convertir le code SQL qui interagit avec les bases de données que vous avez exclues du périmètre de migration. Vous pouvez exclure les fichiers source de votre application C# du champ de conversion. Pour ce faire, décochez les cases correspondant aux noms des fichiers que vous souhaitez exclure de la zone de conversion.

Après avoir modifié l'étendue de conversion, l'analyse du code SQL de tous les fichiers source de vos applications C# se poursuit. AWS SCT copie ensuite dans le dossier cible tous les fichiers source que vous avez exclus de la zone de conversion. Cette opération permet de créer votre application après avoir enregistré les fichiers d'application convertis.

## Enregistrez le code de votre application converti avec AWS SCT

Pour enregistrer le code d'application converti, procédez comme suit.

Pour enregistrer le code d'application converti

1. Développez le nœud C# sous Applications dans le panneau de base de données cible.
2. Choisissez votre application convertie, puis cliquez sur Enregistrer.
3. Entrez le chemin d'accès au dossier pour enregistrer le code d'application converti, puis choisissez Sélectionner un dossier.

## Gestion des projets de conversion d'applications C# dans AWS SCT

Vous pouvez ajouter plusieurs projets de conversion d'applications C#, mettre à jour le code d'application dans le AWS SCT projet ou supprimer un projet de conversion C# de votre AWS SCT projet.

Pour ajouter un projet de conversion d'application C# supplémentaire

1. Développez le nœud Applications dans le panneau de gauche.
2. Choisissez le nœud C# et ouvrez le menu contextuel (clic droit).
3. Choisissez New application (Nouvelle application).
4. Entrez les informations requises pour créer un nouveau projet de conversion d'application C#. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications C#](#).

Après avoir modifié le code source de votre application, chargez-le dans le AWS SCT projet.

Pour télécharger le code d'application mis à jour

1. Développez le nœud C# sous Applications dans le panneau de gauche.
2. Choisissez l'application à mettre à jour, puis ouvrez le menu contextuel (clic droit).



### 3. Choisissez Actualiser, puis Oui.

AWS SCT télécharge le code de votre application depuis les fichiers source et supprime les résultats de conversion. Pour conserver les modifications de code que vous avez apportées AWS SCT et les résultats de la conversion, créez un nouveau projet de conversion en C#.

Pour supprimer un projet de conversion d'application C#

1. Développez le nœud C# sous Applications dans le panneau de gauche.
2. Choisissez l'application à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer, puis OK.

## Création d'un rapport d'évaluation de la conversion des applications C# dans AWS SCT

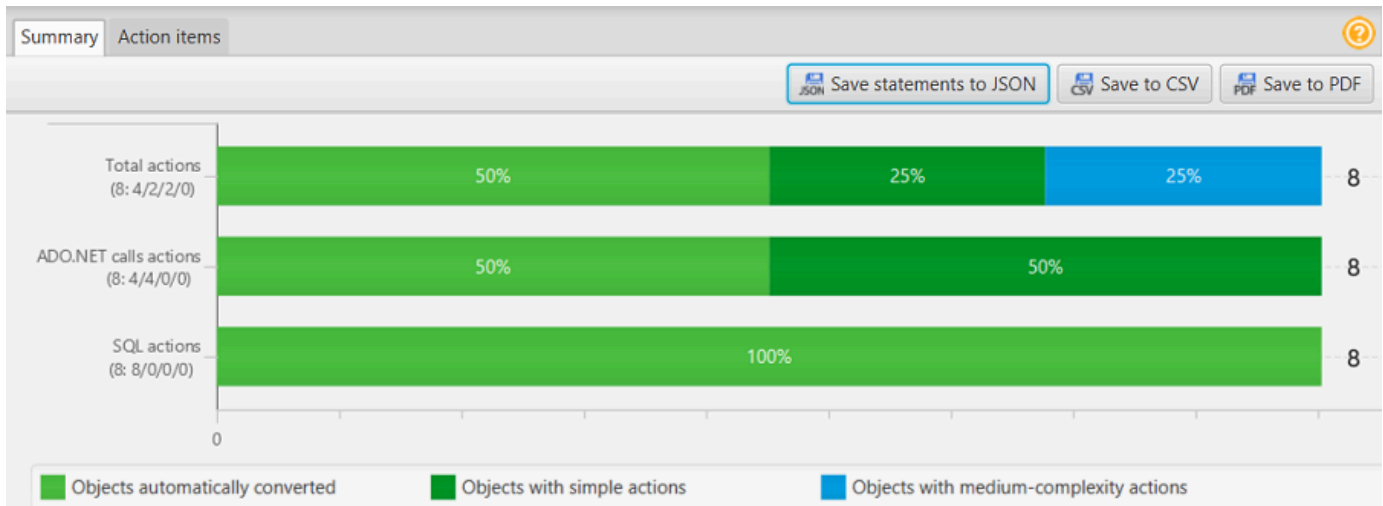
Le rapport d'évaluation de la conversion des applications C# fournit des informations sur la conversion du code SQL intégré à votre application C# en un format compatible avec votre base de données cible. Le rapport d'évaluation fournit les détails de conversion pour tous les points d'exécution SQL et tous les fichiers de code source. Le rapport d'évaluation inclut également des actions pour le code SQL qui ne AWS SCT peut pas être converti.

Utilisez la procédure suivante pour créer un rapport d'évaluation de la conversion des applications C#.

Pour créer un rapport d'évaluation de la conversion des applications C#

1. Développez le nœud C# sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir.
4. Dans le menu Affichage, choisissez Affichage du rapport d'évaluation.
5. Consultez l'onglet Résumé.

L'onglet Résumé, illustré ci-dessous, affiche les informations de synthèse du rapport d'évaluation de l'application C#. Il affiche les résultats de conversion pour tous les points d'exécution SQL et tous les fichiers de code source.



6. Choisissez Enregistrer les instructions au format JSON pour enregistrer le code SQL extrait de votre application C# sous forme de fichier JSON.
7. (Facultatif) Enregistrez une copie locale du rapport sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :

- Choisissez Enregistrer au format PDF en haut à droite pour enregistrer le rapport sous forme de fichier PDF.

Le fichier PDF contient le résumé, les mesures à prendre et les recommandations relatives à la conversion des applications.

- Choisissez Enregistrer au format CSV en haut à droite pour enregistrer le rapport sous forme de fichier CSV.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir le code SQL.

## Conversion de code SQL dans des applications C++ avec AWS Schema Conversion Tool

Pour une conversion d'Oracle vers PostgreSQL, vous pouvez l' AWS SCT utiliser pour convertir le code SQL intégré à vos applications C++. Ce convertisseur d'application C++ spécifique comprend la logique de l'application. Il collecte des instructions situées dans différents objets d'application, tels que des fonctions, des paramètres, des variables locales, etc.

Grâce à cette analyse approfondie, le convertisseur de code SQL de l'application C++ fournit de meilleurs résultats de conversion que le convertisseur générique.

## Création de projets de conversion d'applications C++ dans AWS SCT

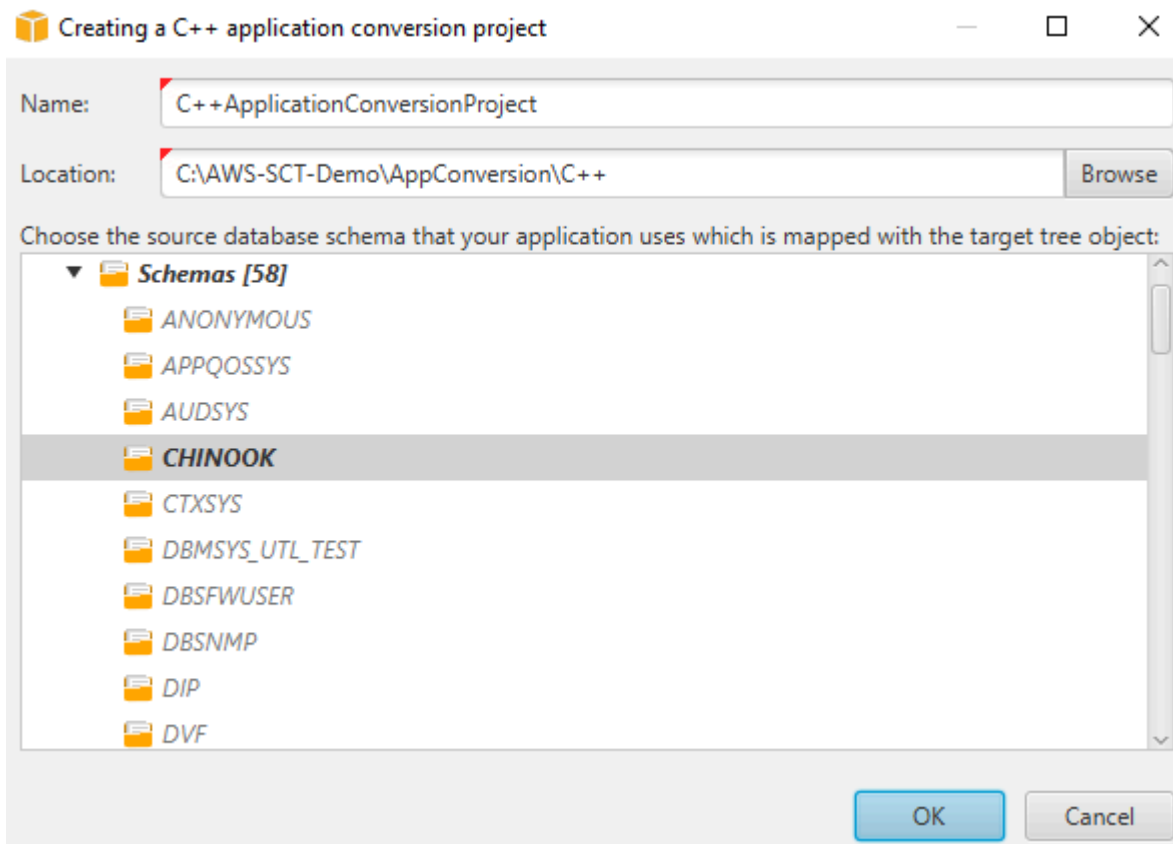
Vous pouvez créer un projet de conversion d'application C++ uniquement pour convertir des schémas de base de données Oracle en schémas de base de données PostgreSQL. Assurez-vous d'ajouter une règle de mappage dans votre projet qui inclut un schéma Oracle source et une base de données PostgreSQL cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Vous pouvez ajouter plusieurs projets de conversion d'applications dans un seul AWS SCT projet.

Pour créer un projet de conversion d'applications C++

1. Créez un projet de conversion de base de données et ajoutez une base de données Oracle source. Pour plus d'informations, consultez [Démarrage et gestion de projets dans AWS SCT](#) et [Ajouter des serveurs au projet dans AWS SCT](#).
2. Ajoutez une règle de mappage qui inclut votre base de données Oracle source et une base de données PostgreSQL cible. Vous pouvez ajouter une base de données PostgreSQL cible ou utiliser une plate-forme de base de données cible PostgreSQL virtuelle dans une règle de mappage. Pour plus d'informations, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#) et [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).
3. Dans le menu Affichage, choisissez Vue principale.
4. Dans le menu Applications, choisissez Nouvelle application C++.

La boîte de dialogue Création d'un projet de conversion d'applications C++ apparaît.



5. Dans Nom, entrez le nom de votre projet de conversion d'application C++. Chaque projet de conversion de schéma de base de données pouvant comporter un ou plusieurs projets de conversion d'applications secondaires, choisissez un nom pertinent si vous ajoutez plusieurs projets.
6. Dans Emplacement, entrez l'emplacement du code source de votre application.
7. Dans l'arborescence des sources, choisissez le schéma utilisé par votre application. Assurez-vous que ce schéma fait partie d'une règle de mappage. AWS SCT met en évidence les schémas qui font partie d'une règle de mappage en gras.
8. Cliquez sur OK pour créer votre projet de conversion d'application C++.
9. Trouvez votre projet de conversion d'applications C++ dans le nœud Applications du panneau de gauche.

## Conversion du code SQL de votre application C++ en AWS SCT

Après avoir ajouté votre application C++ au AWS SCT projet, convertissez le code SQL de cette application dans un format compatible avec votre plate-forme de base de données cible. Utilisez la

procédure suivante pour analyser et convertir le code SQL intégré à votre application C++ dans AWS SCT.

Pour convertir votre code SQL

1. Développez le nœud C++ sous Applications dans le panneau de gauche et choisissez l'application à convertir.
2. Dans le projet d'application Source Oracle, choisissez Settings. Vérifiez et modifiez les paramètres de conversion pour l'application C++ sélectionnée. Vous pouvez également définir les paramètres de conversion pour toutes les applications C++ que vous avez ajoutées à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Gestion des projets de conversion d'applications C++](#).
3. Pour le type de compilateur, choisissez le compilateur que vous utilisez pour le code source de votre application C++. AWS SCT prend en charge les compilateurs C++ suivants : Microsoft Visual C++, GCC, la collection de compilateurs GNU et Clang. L'option par défaut est Microsoft Visual C++.
4. Pour les macros définies par l'utilisateur, entrez le chemin du fichier qui inclut les macros définies par l'utilisateur issues de votre projet C++. Assurez-vous que ce fichier a la structure suivante :`#define name value`. Dans l'exemple précédent, `value` il s'agit d'un paramètre facultatif. La valeur par défaut de ce paramètre facultatif est `1`.

Pour créer ce fichier, ouvrez votre projet dans Microsoft Visual Studio, puis choisissez Projet, Propriétés, C/C++ et Preprocessor. Pour les définitions du préprocesseur, choisissez Modifier et copiez les noms et les valeurs dans un nouveau fichier texte. Ensuite, pour chaque chaîne du fichier, ajoutez le préfixe suivant :`#define` .

5. Pour les répertoires d'inclusion externes, entrez les chemins d'accès aux dossiers contenant les bibliothèques externes que vous utilisez dans votre projet C++.
6. Dans le volet de gauche, choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
7. Choisissez Convertir. AWS SCT analyse vos fichiers de code source, détermine la logique de l'application et charge les métadonnées du code dans le projet. Ces métadonnées de code incluent les classes C++, les objets, les méthodes, les variables globales, les interfaces, etc.

Dans le panneau de base de données cible, AWS SCT crée une structure de dossiers similaire à celle de votre projet d'application source. Vous pouvez consulter ici le code d'application converti, comme indiqué ci-dessous.

The screenshot displays two windows from the AWS Schema Conversion Tool. The top window, titled 'Source Oracle file: StringInitialization.cpp', shows the original C++ code. The bottom window, titled 'Target Amazon RDS for PostgreSQL file: StringInitialization.cpp', shows the converted code. The conversion changes the SQL query from 'SELECT JAVADB.GET\_INT() FROM DUAL' to 'SELECT javadb.get\_int()'.

```

Source Oracle file: StringInitialization.cpp
44  if ((dRet == SQLDriverConnect(hDbc, NULL, lpConnectionStr, connectionStr.size(), OutConnStr, 0xFF
45  {
46      SQLHANDLE hSelectStm = NULL;
47
48      if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hSelectStm)) == SQL_SUCCESS)
49      {
50
51          char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
52          strncpy_s(&buff[0], 0xFF, "SELECT JAVADB.GET_INT() FROM DUAL", 18);
53
54          if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
55          {
56
Target Amazon RDS for PostgreSQL file: StringInitialization.cpp
45  if ((dRet == SQLDriverConnect(hDbc, NULL, lpConnectionStr, connectionStr.size(), OutConnStr,
46  {
47      SQLHANDLE hSelectStm = NULL;
48
49      if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hSelectStm)) == SQL_SUCCESS)
50      {
51
52          char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
53          strncpy_s(&buff[0], 0xFF, "SELECT javadb.get_int()", 18);
54
55          if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
56          {

```

8. Enregistrez le code d'application converti. Pour de plus amples informations, veuillez consulter [Sauvegarde du code d'application converti](#).

## Enregistrez le code de votre application converti avec AWS SCT

Pour enregistrer le code d'application converti, procédez comme suit.

Pour enregistrer le code d'application converti

1. Développez le nœud C++ sous Applications dans le panneau de base de données cible.
2. Choisissez votre application convertie, puis cliquez sur Enregistrer.
3. Entrez le chemin d'accès au dossier pour enregistrer le code d'application converti, puis choisissez Sélectionner un dossier.

## Gestion des projets de conversion d'applications C++ dans AWS SCT

Vous pouvez ajouter plusieurs projets de conversion d'applications C++, modifier les paramètres de conversion, mettre à jour le code d'application C++ ou supprimer un projet de conversion C++ de votre AWS SCT projet.

Pour ajouter un projet de conversion d'applications C++ supplémentaire

1. Développez le nœud Applications dans le panneau de gauche.
2. Choisissez le nœud C++ et ouvrez le menu contextuel (clic droit).
3. Choisissez New application (Nouvelle application).
4. Entrez les informations requises pour créer un nouveau projet de conversion d'application C++. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications C++](#).

Vous pouvez définir les paramètres de conversion pour tous les projets de conversion d'applications C++ de votre AWS SCT projet.

Pour modifier les paramètres de conversion pour toutes les applications C++

1. Dans le menu Paramètres, choisissez Paramètres du projet, puis sélectionnez Conversion d'applications.
2. Pour le type de compilateur, choisissez le compilateur que vous utilisez pour le code source de votre application C++. AWS SCT prend en charge les compilateurs C++ suivants : Microsoft Visual C++, GCC, la collection de compilateurs GNU et Clang. L'option par défaut est Microsoft Visual C++.
3. Pour les macros définies par l'utilisateur, entrez le chemin du fichier qui inclut les macros définies par l'utilisateur issues de votre projet C++. Assurez-vous que ce fichier a la structure suivante :`#define name value`. Dans l'exemple précédent, `value` il s'agit d'un paramètre facultatif. La valeur par défaut de ce paramètre facultatif est `1`.

Pour créer ce fichier, ouvrez votre projet dans Microsoft Visual Studio, puis choisissez Projet, Propriétés, C/C++ et Preprocessor. Pour les définitions du préprocesseur, choisissez Modifier et copiez les noms et les valeurs dans un nouveau fichier texte. Ensuite, pour chaque chaîne du fichier, ajoutez le préfixe suivant `#define` .

4. Pour les répertoires d'inclusion externes, entrez les chemins d'accès aux dossiers contenant les bibliothèques externes que vous utilisez dans votre projet C++.

5. Cliquez sur OK pour enregistrer les paramètres du projet et fermer la fenêtre.

Vous pouvez également définir des paramètres de conversion pour chaque projet de conversion d'applications C++. Pour de plus amples informations, veuillez consulter [Conversion du code SQL de votre application C++](#).

Après avoir modifié le code source de votre application, chargez-le dans le AWS SCT projet.

Pour télécharger le code d'application mis à jour

1. Développez le nœud C++ sous Applications dans le panneau de gauche.
2. Choisissez l'application à mettre à jour, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Actualiser, puis Oui.

AWS SCT télécharge le code de votre application depuis les fichiers source et supprime les résultats de conversion. Pour conserver les modifications de code que vous avez apportées AWS SCT et les résultats de la conversion, créez un nouveau projet de conversion C++.

AWS SCT Supprime également les paramètres de conversion d'application que vous avez spécifiés pour l'application sélectionnée. Après avoir chargé le code d'application mis à jour, AWS SCT applique les valeurs par défaut issues des paramètres du projet.

Pour supprimer un projet de conversion d'applications C++

1. Développez le nœud C++ sous Applications dans le panneau de gauche.
2. Choisissez l'application à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer, puis OK.

## Création d'un rapport d'évaluation de la conversion des applications C++ dans AWS SCT

Le rapport d'évaluation de la conversion des applications C++ fournit des informations sur la conversion du code SQL intégré à votre application C++ en un format compatible avec votre base de données cible. Le rapport d'évaluation fournit les détails de conversion pour tous les points d'exécution SQL et tous les fichiers de code source. Le rapport d'évaluation inclut également des actions pour le code SQL qui ne AWS SCT peut pas être converti.



## Pour créer un rapport d'évaluation de la conversion des applications C++

1. Développez le nœud C++ sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir.
4. Dans le menu Affichage, choisissez Affichage du rapport d'évaluation.
5. Consultez l'onglet Résumé.

L'onglet Résumé affiche les informations de synthèse du rapport d'évaluation de l'application C++. Il affiche les résultats de conversion pour tous les points d'exécution SQL et tous les fichiers de code source.

6. Choisissez Enregistrer les instructions au format JSON pour enregistrer le code SQL extrait de votre application Java sous forme de fichier JSON.
7. (Facultatif) Enregistrez une copie locale du rapport sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :

- Choisissez Enregistrer au format PDF en haut à droite pour enregistrer le rapport sous forme de fichier PDF.

Le fichier PDF contient le résumé, les mesures à prendre et les recommandations relatives à la conversion des applications.

- Choisissez Enregistrer au format CSV en haut à droite pour enregistrer le rapport sous forme de fichier CSV.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir le code SQL.

## Conversion de code SQL dans des applications Java avec AWS Schema Conversion Tool

Pour une conversion d'Oracle vers PostgreSQL, vous pouvez l' AWS Schema Conversion Tool utiliser pour convertir le code SQL intégré à vos applications Java. Ce convertisseur d'application Java spécifique comprend la logique de l'application. Il collecte des instructions situées dans différents objets d'application, tels que des fonctions, des paramètres, des variables locales, etc.

Grâce à cette analyse approfondie, le convertisseur de code SQL de l'application Java fournit de meilleurs résultats de conversion par rapport au convertisseur générique.

Si votre application Java utilise le MyBatis framework pour interagir avec les bases de données, vous pouvez l'utiliser AWS SCT pour convertir des instructions SQL incorporées dans des fichiers MyBatis XML et des annotations. Pour comprendre la logique de ces instructions SQL, utilisez AWS SCT le fichier MyBatis de configuration. AWS SCT peut détecter automatiquement ce fichier dans le dossier de votre application, ou vous pouvez saisir le chemin d'accès à ce fichier manuellement.

## Création de projets de conversion d'applications Java dans AWS SCT

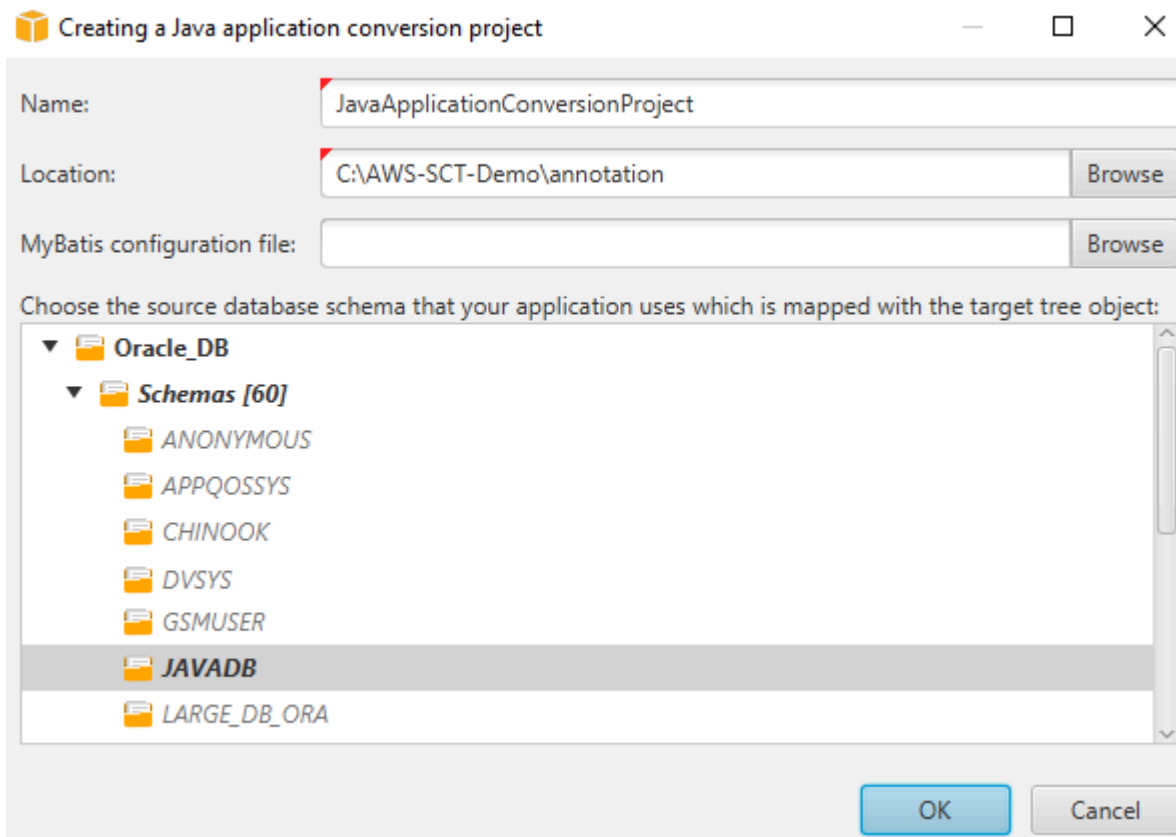
Vous pouvez créer un projet de conversion d'application Java uniquement pour convertir des schémas de base de données Oracle en schémas de base de données PostgreSQL. Assurez-vous d'ajouter une règle de mappage dans votre projet qui inclut un schéma Oracle source et une base de données PostgreSQL cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Vous pouvez ajouter plusieurs projets de conversion d'applications dans un seul AWS SCT projet. Utilisez la procédure suivante pour créer un projet de conversion d'application Java.

Pour créer un projet de conversion d'applications Java

1. Créez un projet de conversion de base de données et ajoutez une base de données Oracle source. Pour plus d'informations, consultez [Démarrage et gestion de projets dans AWS SCT](#) et [Ajouter des serveurs au projet dans AWS SCT](#).
2. Ajoutez une règle de mappage qui inclut votre base de données Oracle source et une base de données PostgreSQL cible. Vous pouvez ajouter une base de données PostgreSQL cible ou utiliser une plate-forme de base de données cible PostgreSQL virtuelle dans une règle de mappage. Pour plus d'informations, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#) et [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).
3. Dans le menu Affichage, choisissez Vue principale.
4. Dans le menu Applications, choisissez Nouvelle application Java.

La boîte de dialogue Création d'un projet de conversion d'applications Java apparaît.



5. Dans Nom, entrez le nom de votre projet de conversion d'application Java. Chaque projet de conversion de schéma de base de données pouvant comporter un ou plusieurs projets de conversion d'applications secondaires, choisissez un nom pertinent si vous ajoutez plusieurs projets.
6. Dans Emplacement, entrez l'emplacement du code source de votre application.
7. (Facultatif) Pour le fichier de MyBatis configuration, entrez le chemin d'accès au fichier MyBatis de configuration. AWS SCT analyse le dossier de votre application pour découvrir automatiquement ce fichier. Si ce fichier ne se trouve pas dans le dossier de votre application ou si vous utilisez plusieurs fichiers de configuration, entrez le chemin manuellement.
8. Dans l'arborescence des sources, choisissez le schéma utilisé par votre application. Assurez-vous que ce schéma fait partie d'une règle de mappage. AWS SCT met en évidence les schémas qui font partie d'une règle de mappage en gras.
9. Cliquez sur OK pour créer votre projet de conversion d'application Java.
10. Trouvez votre projet de conversion d'applications Java dans le nœud Applications du panneau de gauche.

## Conversion du code SQL de votre application Java en AWS SCT

Après avoir ajouté votre application Java au AWS SCT projet, convertissez le code SQL de cette application dans un format compatible avec votre plate-forme de base de données cible. Utilisez la procédure suivante pour analyser et convertir le code SQL intégré à votre application Java dans le AWS Schema Conversion Tool.

Pour convertir votre code SQL

1. Développez le nœud Java sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir. AWS SCT analyse vos fichiers de code source, détermine la logique de l'application et charge les métadonnées du code dans le projet. Ces métadonnées de code incluent les classes Java, les objets, les méthodes, les variables globales, les interfaces, etc.

Dans le panneau de base de données cible, AWS SCT crée une structure de dossiers similaire à celle de votre projet d'application source. Vous pouvez consulter ici le code d'application converti.

```
Source Oracle file: CallMethod2.java
13 private final String USER = "min_privs";
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM JAVADB.DATATYPE_MIXED_AL
26     cs.execute();
27 }
28 }
29 }
```

Cursor position: 697

```
Target Amazon RDS for PostgreSQL file: CallMethod2.java
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26     cs.execute();
27 }
28 }
29 }
```

4. Enregistrez le code d'application converti. Pour de plus amples informations, veuillez consulter [Sauvegarde du code d'application converti](#).

Vos applications Java peuvent inclure du code SQL qui interagit avec différentes bases de données sources. Vous pouvez migrer vers PostgreSQL plusieurs de ces bases de données sources. Dans ce cas, assurez-vous de ne pas convertir le code SQL qui interagit avec les bases de données que vous avez exclues du périmètre de migration. Vous pouvez exclure les fichiers source de votre application Java de la zone de conversion. Pour ce faire, décochez les cases correspondant aux noms des fichiers que vous souhaitez exclure de la zone de conversion.

Après avoir modifié l'étendue de conversion, AWS SCT l'analyse du code SQL de tous les fichiers source de vos applications Java se poursuit. AWS SCT Copie ensuite dans le dossier cible tous les fichiers source que vous avez exclus de la zone de conversion. Cette opération permet de créer votre application après avoir enregistré les fichiers d'application convertis.

## Enregistrez le code de votre application converti avec AWS SCT

Pour enregistrer le code d'application converti, procédez comme suit.

Pour enregistrer le code d'application converti

1. Développez le nœud Java sous Applications dans le panneau de base de données cible.
2. Choisissez votre application convertie, puis cliquez sur Enregistrer.
3. Entrez le chemin d'accès au dossier pour enregistrer le code d'application converti, puis choisissez Sélectionner un dossier.

Si votre application Java source utilise le MyBatis framework, assurez-vous de mettre à jour votre fichier de configuration pour qu'il fonctionne avec votre nouvelle base de données.

## Gestion des projets de conversion d'applications Java dans AWS SCT

Vous pouvez ajouter plusieurs projets de conversion d'applications Java, mettre à jour le code de l'application dans le AWS SCT projet ou supprimer un projet de conversion Java de votre AWS SCT projet.

Pour ajouter un projet de conversion d'application Java supplémentaire

1. Développez le nœud Applications dans le panneau de gauche.
2. Choisissez le nœud Java et ouvrez le menu contextuel (clic droit).
3. Choisissez New application (Nouvelle application).
4. Entrez les informations requises pour créer un nouveau projet de conversion d'applications Java. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications Java](#).

Après avoir modifié le code source de votre application, chargez-le dans le AWS SCT projet.

Pour télécharger le code d'application mis à jour

1. Développez le nœud Java sous Applications dans le panneau de gauche.
2. Choisissez l'application à mettre à jour, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Actualiser, puis Oui.

AWS SCT télécharge le code de votre application depuis les fichiers source et supprime les résultats de conversion. Pour conserver les modifications de code que vous avez apportées AWS SCT et les résultats de la conversion, créez un nouveau projet de conversion Java.

Si votre application Java source utilise le MyBatis framework, AWS SCT utilise le fichier MyBatis de configuration pour analyser votre code SQL. Après avoir modifié ce fichier, chargez-le dans le AWS SCT projet.

Pour modifier le chemin d'accès au fichier MyBatis de configuration

1. Développez le nœud Java sous Applications dans le panneau de gauche.
2. Choisissez votre application, puis sélectionnez Paramètres.
3. Choisissez Parcourir, puis choisissez le fichier MyBatis de configuration.
4. Choisissez Appliquer.
5. Dans le panneau de gauche, choisissez votre application, ouvrez le menu contextuel (clic droit) et choisissez Actualiser.

Pour supprimer un projet de conversion d'applications Java

1. Développez le nœud Java sous Applications dans le panneau de gauche.
2. Choisissez l'application à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer, puis OK.

## Création d'un rapport d'évaluation de la conversion d'une application Java dans AWS SCT

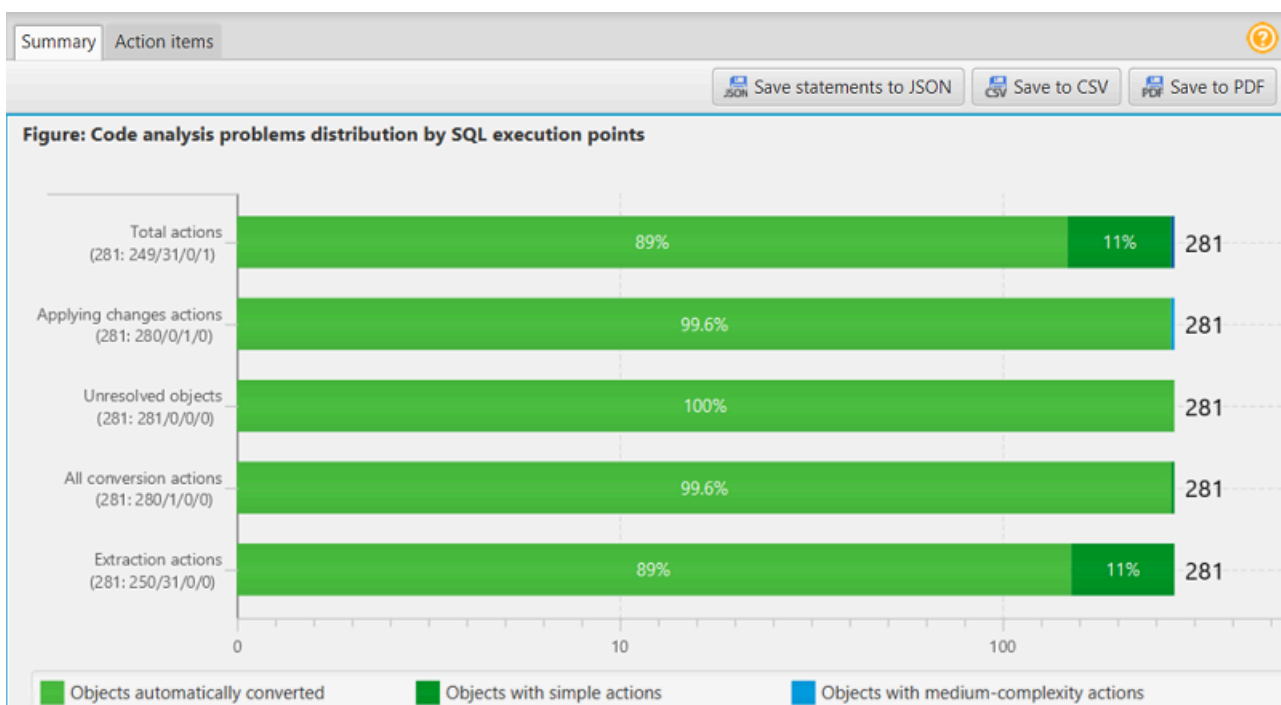
Le rapport d'évaluation de la conversion des applications Java fournit des informations sur la conversion du code SQL intégré à votre application Java en un format compatible avec votre base de données cible. Le rapport d'évaluation fournit les détails de conversion pour tous les points d'exécution SQL et tous les fichiers de code source. Le rapport d'évaluation inclut également des actions pour le code SQL que AWS SCT ne peut pas convertir.

Utilisez la procédure suivante pour créer un rapport d'évaluation de la conversion des applications Java.

## Pour créer un rapport d'évaluation de la conversion des applications Java

1. Développez le nœud Java sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir.
4. Dans le menu Affichage, choisissez Affichage du rapport d'évaluation.
5. Vérifiez l'onglet Summary.

L'onglet Résumé, illustré ci-dessous, affiche les informations de synthèse du rapport d'évaluation de l'application Java. Il affiche les résultats de conversion pour tous les points d'exécution SQL et tous les fichiers de code source.



6. Choisissez Enregistrer les instructions au format JSON pour enregistrer le code SQL extrait de votre application Java sous forme de fichier JSON.
7. (Facultatif) Enregistrez une copie locale du rapport sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :
  - Choisissez Enregistrer au format PDF en haut à droite pour enregistrer le rapport sous forme de fichier PDF.

Le fichier PDF contient le résumé, les mesures à prendre et les recommandations relatives à la conversion des applications.



- Choisissez Enregistrer au format CSV en haut à droite pour enregistrer le rapport sous forme de fichier CSV.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir le code SQL.

## Conversion de code SQL dans les applications Pro\*C avec AWS Schema Conversion Tool

Pour une conversion d'Oracle vers PostgreSQL, vous pouvez utiliser AWS Schema Conversion Tool le AWS SCT() pour convertir le code SQL intégré à vos applications Pro\*C. Ce convertisseur d'application Pro\*C spécifique comprend la logique de l'application. Il collecte des instructions situées dans différents objets d'application, tels que des fonctions, des paramètres, des variables locales, etc.

Grâce à cette analyse approfondie, le convertisseur de code SQL de l'application Pro\*C fournit de meilleurs résultats de conversion par rapport au convertisseur générique.

### Création de projets de conversion d'applications Pro\*C dans AWS SCT

Vous pouvez créer un projet de conversion d'application Pro\*C uniquement pour convertir des schémas de base de données Oracle en schémas de base de données PostgreSQL. Assurez-vous d'ajouter une règle de mappage dans votre projet qui inclut un schéma Oracle source et une base de données PostgreSQL cible. Pour de plus amples informations, veuillez consulter [Cartographie des types de données dans AWS Schema Conversion Tool](#).

Vous pouvez ajouter plusieurs projets de conversion d'applications dans un seul AWS SCT projet. Utilisez la procédure suivante pour créer un projet de conversion d'application Pro\*C.

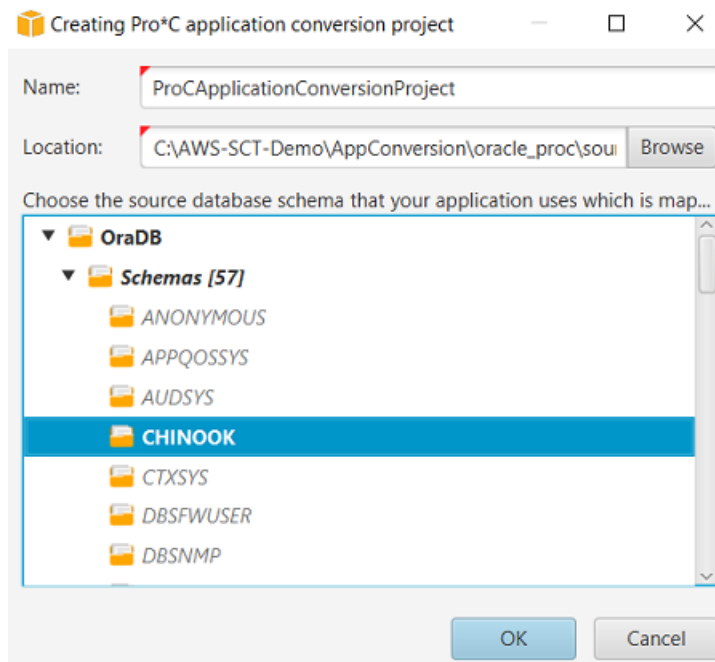
Pour créer un projet de conversion d'applications Pro\*C

1. Créez un projet de conversion de base de données et ajoutez une base de données Oracle source. Pour plus d'informations, consultez [Démarrage et gestion de projets dans AWS SCT](#) et [Ajouter des serveurs au projet dans AWS SCT](#).
2. Ajoutez une règle de mappage qui inclut votre base de données Oracle source et une base de données PostgreSQL cible. Vous pouvez ajouter une base de données PostgreSQL cible ou utiliser une plate-forme de base de données cible PostgreSQL virtuelle dans une règle de

mappage. Pour plus d'informations, consultez [Cartographie des types de données dans AWS Schema Conversion Tool](#) et [Mappage vers des cibles virtuelles dans AWS Schema Conversion Tool](#).

3. Dans le menu Affichage, choisissez Vue principale.
4. Dans le menu Applications, choisissez Nouvelle application Pro\*C.

La boîte de dialogue Création d'un projet de conversion d'applications Pro\*C apparaît.



5. Dans Nom, entrez le nom de votre projet de conversion d'applications Pro\*C. Chaque projet de conversion de schéma de base de données pouvant comporter un ou plusieurs projets de conversion d'applications secondaires, choisissez un nom pertinent si vous ajoutez plusieurs projets.
6. Dans Emplacement, entrez l'emplacement du code source de votre application.
7. Dans l'arborescence des sources, choisissez le schéma utilisé par votre application. Assurez-vous que ce schéma fait partie d'une règle de mappage. AWS SCT met en évidence les schémas qui font partie d'une règle de mappage en gras.
8. Cliquez sur OK pour créer votre projet de conversion d'application Pro\*C.
9. Trouvez votre projet de conversion d'applications Pro\*C dans le nœud Applications du panneau de gauche.

## Conversion du code SQL de votre application Pro\*C en AWS SCT

Après avoir ajouté votre application Pro\*C au AWS SCT projet, convertissez le code SQL de cette application dans un format compatible avec votre plate-forme de base de données cible. Utilisez la procédure suivante pour analyser et convertir le code SQL intégré à votre application Pro\*C dans le AWS Schema Conversion Tool

Pour convertir votre code SQL

1. Développez le nœud Pro\*C sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir, puis sélectionnez Paramètres.
  - a. Pour le chemin du fichier d'en-tête global, entrez le chemin des fichiers d'en-tête utilisés par votre projet d'application.
  - b. Choisissez Interpréter toutes les variables hôtes non résolues comme pour voir toutes les variables non résolues dans le code converti.
  - c. Choisissez Utiliser la fonction de conversion de chaînes à largeur fixe dans le pack d'extensions pour utiliser les fonctions du pack d'extensions dans le code SQL converti. AWS SCT inclut les fichiers du pack d'extension dans votre projet d'application.
  - d. Choisissez Transformer les blocs PL/SQL anonymes en appels SQL autonomes ou en fonctions stockées afin de créer des procédures stockées dans votre base de données cible pour tous les blocs PL/SQL anonymes. AWS SCT inclut ensuite les exécutions de ces procédures stockées dans le code d'application converti.
  - e. Choisissez Utiliser un flux de curseur personnalisé pour améliorer la conversion des curseurs de base de données Oracle.
3. Dans le panneau de gauche, choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
4. Choisissez Convertir. AWS SCT analyse vos fichiers de code source, détermine la logique de l'application et charge les métadonnées du code dans le projet. Ces métadonnées de code incluent les classes Pro\*C, les objets, les méthodes, les variables globales, les interfaces, etc.

Dans le panneau de base de données cible, AWS SCT crée une structure de dossiers similaire à celle de votre projet d'application source. Vous pouvez consulter ici le code d'application converti.

The screenshot displays two panels of the AWS Schema Conversion Tool. The top panel, titled 'Source Oracle app function: main() int', shows the original Oracle Pro\*C code with line numbers 11 through 20. The code includes a variable declaration, a comment, and SQL execution statements. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted PostgreSQL SQL code with line numbers 8 through 18. The converted code uses 'EXEC SQL' to execute the same logic as the original Pro\*C code.

```
Source Oracle app function: main() int
Properties Text Related converted objects Statistics Settings
11 int i = 5;
12
13 /* Connect string */
14
15 EXEC SQL INSERT INTO embeddedc.t_insert(i) VALUES (:i);
16
17 EXEC SQL COMMIT;
18
19 return 0;
20 }
Cursor position: 118 Click position: 197

Target Amazon RDS for PostgreSQL app function: main() int
Properties Text Apply status Key management
8
9 EXEC SQL int i = 5;
10
11 /* Connect string */
12
13 EXEC SQL INSERT INTO embeddedc.t_insert (i)
14 VALUES (:i);
15
16 EXEC SQL COMMIT;
17
18 return 0;
```

5. Enregistrez le code d'application converti. Pour de plus amples informations, veuillez consulter [Modification et enregistrement du code d'application converti](#).

## Modifier et enregistrer le code de votre application converti avec AWS SCT

Vous pouvez modifier les instructions SQL converties et les utiliser AWS SCT pour intégrer ce code modifié dans le code d'application Pro\*C converti. Pour modifier le code SQL converti, procédez comme suit.

Pour modifier votre code SQL converti

1. Développez le nœud Pro\*C sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir, ouvrez le menu contextuel (clic droit), puis choisissez Convertir.
3. Dans le menu Affichage, choisissez Affichage du rapport d'évaluation.
4. Choisissez Enregistrer les instructions au format CSV pour enregistrer le code SQL extrait de votre application Pro\*C sous forme de fichier CSV.
5. Entrez le nom du fichier CSV pour enregistrer le code SQL extrait, puis choisissez Enregistrer.
6. Modifiez le code SQL extrait.
7. Dans le menu Affichage, choisissez Vue principale.

8. Développez le nœud Pro\*C sous Applications dans le panneau de base de données cible.
9. Choisissez votre application convertie, ouvrez le menu contextuel (clic droit) et choisissez Importer des relevés depuis un fichier CSV.
10. Choisissez Oui, puis choisissez le fichier contenant votre code SQL modifié, puis choisissez Ouvrir.

AWS SCT divise les instructions SQL converties en plusieurs parties et les place dans les objets appropriés du code source de votre application. Pour enregistrer le code d'application converti, procédez comme suit.

Pour enregistrer le code d'application converti

1. Développez le nœud Pro\*C sous Applications dans le panneau de base de données cible.
2. Choisissez votre application convertie, puis cliquez sur Enregistrer.
3. Entrez le chemin d'accès au dossier pour enregistrer le code d'application converti, puis choisissez Sélectionner un dossier.

## Gestion des projets de conversion d'applications Pro\*C dans AWS SCT

Vous pouvez ajouter plusieurs projets de conversion d'applications Pro\*C, mettre à jour le code d'application dans le AWS SCT projet ou supprimer un projet de conversion Pro\*C de votre projet.

### AWS SCT

Pour ajouter un projet de conversion d'applications Pro\*C supplémentaire

1. Développez le nœud Applications dans le panneau de gauche.
2. Choisissez le nœud Pro\*C et ouvrez le menu contextuel (clic droit).
3. Choisissez New application (Nouvelle application).
4. Entrez les informations requises pour créer un nouveau projet de conversion d'applications Pro\*C. Pour de plus amples informations, veuillez consulter [Création de projets de conversion d'applications Pro\\*C](#).

Après avoir modifié le code source de votre application, chargez-le dans le AWS SCT projet.

## Pour télécharger le code d'application mis à jour

1. Développez le nœud Pro\*C sous Applications dans le panneau de gauche.
2. Choisissez l'application à mettre à jour, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Actualiser, puis Oui.

AWS SCT télécharge le code de votre application depuis les fichiers source et supprime les résultats de conversion. Pour conserver les modifications de code que vous avez apportées AWS SCT et les résultats de la conversion, créez un nouveau projet de conversion Pro\*C.

## Pour supprimer un projet de conversion d'applications Pro\*C

1. Développez le nœud Pro\*C sous Applications dans le panneau de gauche.
2. Choisissez l'application à supprimer, puis ouvrez le menu contextuel (clic droit).
3. Choisissez Supprimer, puis OK.

## Création d'un rapport d'évaluation de la conversion des applications Pro\*C dans AWS SCT

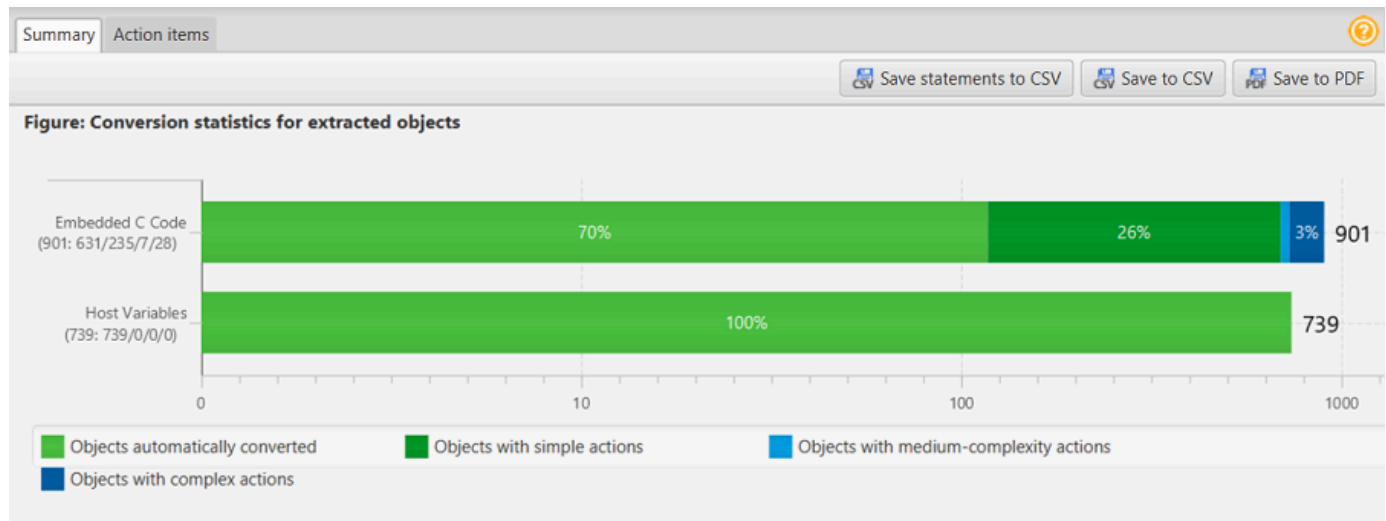
Le rapport d'évaluation de la conversion des applications Pro\*C fournit des informations sur la conversion du code SQL intégré à votre application Pro\*C en un format compatible avec votre base de données cible. Le rapport d'évaluation fournit les détails de conversion pour tous les points d'exécution SQL et tous les fichiers de code source. Le rapport d'évaluation inclut également des actions pour le code SQL que AWS SCT ne peut pas convertir.

Utilisez la procédure suivante pour créer un rapport d'évaluation de la conversion des applications Pro\*C.

### Pour créer un rapport d'évaluation de la conversion des applications Pro\*C

1. Développez le nœud Pro\*C sous Applications dans le panneau de gauche.
2. Choisissez l'application à convertir et ouvrez le menu contextuel (clic droit).
3. Choisissez Convertir.
4. Dans le menu Affichage, choisissez Affichage du rapport d'évaluation.
5. Vérifiez l'onglet Summary.

L'onglet Résumé, illustré ci-dessous, affiche les informations de synthèse du rapport d'évaluation de l'application Pro\*C. Il affiche les résultats de conversion pour tous les points d'exécution SQL et tous les fichiers de code source.



6. Choisissez Enregistrer les instructions au format CSV pour enregistrer le code SQL extrait de votre application Pro\*C sous forme de fichier CSV (valeurs séparées par des virgules).
7. (Facultatif) Enregistrez une copie locale du rapport sous forme de fichier PDF ou de fichier de valeurs séparées par des virgules (CSV) :
  - Choisissez Enregistrer au format PDF en haut à droite pour enregistrer le rapport sous forme de fichier PDF.

Le fichier PDF contient le résumé, les mesures à prendre et les recommandations relatives à la conversion des applications.

- Choisissez Enregistrer au format CSV en haut à droite pour enregistrer le rapport sous forme de fichier CSV.

Le fichier CSV contient des actions, des actions recommandées et une estimation de la complexité de l'effort manuel requis pour convertir le code SQL.

# Utilisation de packs d'extension avec AWS Schema Conversion Tool

Un pack d'AWS SCT extension est un module complémentaire qui émule les fonctions présentes dans une base de données source qui sont requises lors de la conversion d'objets vers la base de données cible. Avant de pouvoir installer un pack d' AWS SCT extension, vous devez convertir le schéma de votre base de données.

Chaque pack d' AWS SCT extension inclut les composants suivants :

- Schéma de base de données : inclut des fonctions, des procédures et des tables SQL permettant d'émuler certains objets de base de données de traitement des transactions en ligne (OLTP) et de traitement analytique en ligne (OLAP), tels que les séquences. Émule également des émulations non prises en charge built-in-fonctions à partir de la base de données source. Le nom de ce schéma a le format suivant : `aws_ database_engine_name _ext`.
- AWS Lambda fonctions (pour certaines bases de données OLTP) : inclut des AWS Lambda fonctions qui émulent des fonctionnalités de base de données complexes, telles que la planification des tâches et l'envoi d'e-mails.
- Bibliothèques personnalisées pour les bases de données OLAP : inclut un ensemble de bibliothèques Java et Python que vous pouvez utiliser pour migrer des scripts d'extraction, de transformation et de chargement (ETL) de Microsoft SQL Server Integration Services (SSIS) vers AWS Glue ou AWS Glue Studio.

Les bibliothèques Java incluent les modules suivants :

- `spark-excel_2.11-0.13.1.jar`— Pour émuler les fonctionnalités des composants source et cible d'Excel.
- `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar`, et `xmlbeans-3.1.0.jar` — Pour émuler les fonctionnalités du composant source XML.

Les bibliothèques Python incluent les modules suivants :

- `sct_utils.py`— Pour émuler les types de données source et préparer les paramètres de la requête SQL Spark.
- `ssis_datetime.py`— Pour émuler les fonctions intégrées de date et d'heure.
- `ssis_null.py`— Pour émuler ISNULL les fonctions REPLACENULL intégrées.
- `ssis_string.py`— Pour émuler les fonctions intégrées aux chaînes.



Pour plus d'informations sur ces bibliothèques, consultez [Utilisation de bibliothèques personnalisées pour les packs d' AWS SCT extension](#).

Vous pouvez appliquer des packs d' AWS SCT extension de deux manières :

- AWS SCT peut appliquer automatiquement un pack d'extension lorsque vous appliquez un script de base de données cible en choisissant Appliquer à la base de données dans le menu contextuel. AWS SCT applique le pack d'extension avant d'appliquer tous les autres objets du schéma.
- Pour appliquer manuellement un pack d'extension, choisissez la base de données cible, puis choisissez Appliquer le pack d'extension pour dans le menu contextuel (clic droit). Dans la plupart des cas, une application automatique est suffisante. Cependant, vous souhaitez peut-être appliquer le pack manuellement s'il est supprimé accidentellement.

Chaque fois que vous appliquez un pack d' AWS SCT extension à un magasin de données cible, les composants sont remplacés et une notification AWS SCT s'affiche à ce sujet. Pour désactiver ces notifications, choisissez Paramètres, Paramètres généraux, Notifications, puis sélectionnez Masquer l'alerte de remplacement du pack d'extension.

Pour effectuer une conversion de Microsoft SQL Server vers PostgreSQL, vous pouvez utiliser le pack d'extension SQL Server vers PostgreSQL dans AWS SCT. Ce pack d'extension émule l'agent SQL Server et le courrier de base de données SQL Server. Pour plus d'informations, consultez [Émulation de l'agent SQL Server dans PostgreSQL avec un pack d'extension](#) et [Émulation du courrier de base de données SQL Server dans PostgreSQL avec un pack d'extension](#).

Vous trouverez ci-dessous plus d'informations sur l'utilisation des packs d' AWS SCT extension.

## Rubriques

- [Autorisations d'utilisation du pack d' AWS SCT extension](#)
- [Utilisation du schéma du pack d'extension](#)
- [Utilisation de bibliothèques personnalisées pour les packs d' AWS SCT extension](#)
- [Utilisation des AWS Lambda fonctions du pack d' AWS SCT extension](#)
- [Configuration des fonctions du pack d' AWS SCT extension](#)

## Autorisations d'utilisation du pack d' AWS SCT extension

Le pack d' AWS SCT extension pour Amazon Aurora émule l'envoi d'e-mails, la planification des tâches, la mise en file d'attente et d'autres opérations à l'aide de fonctions. AWS Lambda Lorsque vous appliquez le pack d' AWS SCT extension à votre base de données Aurora cible, vous AWS SCT créez un nouveau rôle AWS Identity and Access Management (IAM) et une politique IAM intégrée. Ensuite, AWS SCT crée une nouvelle fonction Lambda et configure votre cluster de base de données Aurora pour les connexions sortantes vers. AWS Lambda Pour exécuter ces opérations, assurez-vous d'accorder les autorisations requises suivantes à votre utilisateur IAM :

- `iam:CreateRole`— pour créer un nouveau rôle IAM pour votre AWS compte.
- `iam:CreatePolicy`— pour créer une nouvelle politique IAM pour votre AWS compte.
- `iam:AttachRolePolicy`— pour associer la politique spécifiée à votre rôle IAM.
- `iam:PutRolePolicy`— pour mettre à jour un document de politique intégré à votre rôle IAM.
- `iam:PassRole`— pour transmettre le rôle IAM spécifié au moteur de règles.
- `iam:TagRole`— pour ajouter des balises à un rôle IAM.
- `iam:TagPolicy`— pour ajouter des balises à une politique IAM.
- `lambda:ListFunctions`— pour voir la liste de vos fonctions Lambda.
- `lambda:ListTags`— pour voir la liste des tags de vos fonctions Lambda.
- `lambda:CreateFunction`— pour créer une nouvelle fonction Lambda.
- `rds:AddRoleToDBCluster`— pour associer un rôle IAM à votre cluster de base de données Aurora.

Le pack d' AWS SCT extension pour Amazon Redshift émule les fonctions de base de l'entrepôt de données source requises lors de l'application d'objets convertis à Amazon Redshift. Avant d'appliquer votre code converti à Amazon Redshift, vous devez appliquer le pack d'extension pour Amazon Redshift. Pour ce faire, incluez `iam:SimulatePrincipalPolicyaction` dans votre politique IAM.

AWS SCT utilise le simulateur de politique IAM pour vérifier les autorisations requises pour installer le pack d'extension Amazon Redshift. Le simulateur de politique IAM peut afficher un message d'erreur même si vous avez correctement configuré votre utilisateur IAM. Il s'agit d'un problème connu du simulateur de politique IAM. En outre, le simulateur de politique IAM affiche un message d'erreur lorsque `iam:SimulatePrincipalPolicyaction` n'est pas inscrite dans votre stratégie IAM. Dans ces cas, vous pouvez ignorer le message d'erreur et appliquer le pack d'extension à l'aide de

l'assistant du pack d'extension. Pour de plus amples informations, veuillez consulter [Appliquer le pack d'extension](#).

## Utilisation du schéma du pack d'extension

Lorsque vous convertissez le schéma de votre base de données ou de votre entrepôt de données, AWS SCT ajoute un schéma supplémentaire à votre base de données cible. Ce schéma met en œuvre les fonctions système SQL de la base de données source nécessaires lors de l'écriture du schéma converti dans la base de données cible. Ce schéma supplémentaire est appelé schéma du kit d'extension.

Le schéma du kit d'extension des bases de données OLTP est nommé en fonction de la base de données source, comme suit :

- Microsoft SQL Server: AWS\_SQLSERVER\_EXT
- MySQL: AWS\_MYSQL\_EXT
- Oracle: AWS\_ORACLE\_EXT
- PostgreSQL: AWS\_POSTGRESQL\_EXT

Le schéma du kit d'extension des applications d'entrepôt de données OLAP est nommé en fonction du magasin de données source, comme suit :

- Greenplum : AWS\_GREENPLUM\_EXT
- Microsoft SQL Server: AWS\_SQLSERVER\_EXT
- Netezza : AWS\_NETEZZA\_EXT
- Oracle: AWS\_ORACLE\_EXT
- Teradata: AWS\_TERADATA\_EXT
- Vertica : AWS\_VERTICA\_EXT

## Utilisation de bibliothèques personnalisées pour les packs d' AWS SCT extension

Dans certains cas, il est AWS SCT impossible de convertir les fonctionnalités de base de données source en fonctionnalités équivalentes dans votre base de données cible. Le pack d' AWS

SCT extension correspondant contient des bibliothèques personnalisées qui émulent certaines fonctionnalités de la base de données source dans votre base de données cible.

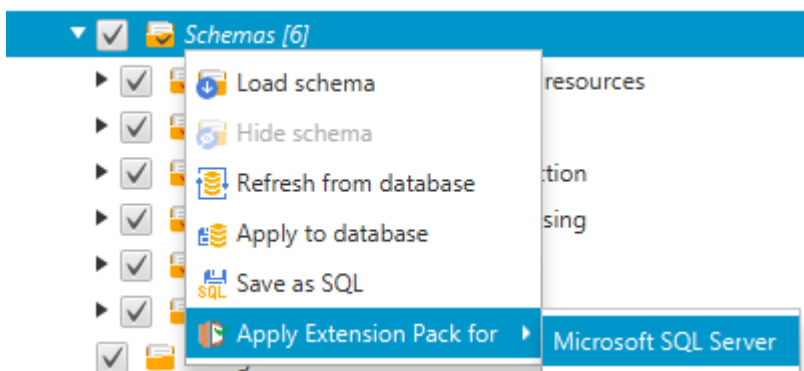
Si vous convertissez une base de données transactionnelle, consultez [Utilisation des AWS Lambda fonctions du pack d' AWS SCT extension](#) .

## Appliquer le pack d'extension

Vous pouvez appliquer le pack d' AWS SCT extension à l'aide de l'assistant du pack d'extension ou lorsque vous appliquez le code converti à votre base de données cible.

Pour appliquer le pack d'extension à l'aide de l'assistant du pack d'extension

1. Dans l' AWS Schema Conversion Tool arborescence de la base de données cible, ouvrez le menu contextuel (clic droit), choisissez Appliquer le pack d'extension pour, puis choisissez votre plate-forme de base de données source.



L'assistant Kit d'extension s'affiche.

2. Lisez la page d'accueil, puis choisissez Next.
3. Sur la page des paramètres du AWS profil, procédez comme suit :
  - Si vous réinstallez uniquement le schéma du kit d'extension, choisissez Skip this step for now, puis Next. L'option Ignorer cette étape pour le moment n'est disponible que pour les bases de données de traitement des transactions en ligne (OLTP).
  - Si vous chargez une nouvelle bibliothèque, fournissez les informations d'identification pour vous connecter à votre Compte AWS. Utilisez cette étape uniquement lorsque vous convertissez des bases de données OLAP ou des scripts ETL. Vous pouvez utiliser vos informations d'identification AWS Command Line Interface (AWS CLI) si vous les avez AWS CLI installées. Vous pouvez également utiliser les informations d'identification précédemment stockées dans un profil dans les paramètres globaux de l'application et associées au projet. Si

nécessaire, choisissez Naviguer vers les paramètres globaux pour configurer ou associer un profil différent à votre AWS SCT projet. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).

4. Si vous chargez une nouvelle bibliothèque, choisissez Je dois télécharger une bibliothèque sur la page de téléchargement de la bibliothèque. Utilisez cette étape uniquement lorsque vous convertissez des bases de données OLAP ou des scripts ETL. Indiquez ensuite le chemin Amazon S3, puis choisissez Upload library to S3.

Si vous avez déjà chargé la bibliothèque, sélectionnez J'ai déjà téléchargé des bibliothèques, utilisez mon compartiment S3 existant sur la page de téléchargement de la bibliothèque. Indiquez ensuite le chemin Amazon S3.

Lorsque vous avez terminé, cliquez sur Next.

5. Sur la page d'émulation de fonctions, choisissez Create extension pack. Des messages affichent l'état des opérations du kit d'extension.

Lorsque vous avez terminé, choisissez Finish.

Pour appliquer le pack d'extension lors de l'application du code converti

1. Spécifiez le compartiment Amazon S3 dans votre profil AWS de service. Utilisez cette étape uniquement lorsque vous convertissez des bases de données OLAP ou des scripts ETL. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).

Assurez-vous que votre politique relative aux compartiments Amazon S3 inclut les autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
```

```
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": ["iam:SimulatePrincipalPolicy"],
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": ["iam:GetUser"],
    "Resource": ["arn:aws:iam::111122223333:user/DataExtractionAgentName"]
  }
]
```

Dans l'exemple précédent, remplacez *111122223333:user/DataExtractionAgentName* par le nom de votre utilisateur IAM.

2. Convertissez les schémas de votre entrepôt de données source. Pour de plus amples informations, veuillez consulter [Conversion de schémas d'entrepôts de données](#).
3. Dans le volet droit, choisissez le schéma converti.
4. Ouvrez le menu contextuel (clic droit) pour l'élément de schéma, puis choisissez Apply to database.
5. AWS SCT génère des packs d'extension avec les composants requis et ajoute le `aws_database_engine_name_ext` schéma dans l'arborescence cible. Ensuite, AWS SCT applique le code converti et le schéma du pack d'extension à votre entrepôt de données cible.

Lorsque vous utilisez à la fois Amazon Redshift et AWS Glue comme plate-forme de base de données cible, vous ajoutez un AWS SCT schéma supplémentaire dans le pack d'extension.

## Utilisation des AWS Lambda fonctions du pack d' AWS SCT extension

AWS SCT fournit un pack d'extension qui contient des fonctions Lambda pour le courrier électronique, la planification des tâches et d'autres fonctionnalités pour les bases de données hébergées sur Amazon. EC2

## Utilisation de AWS Lambda fonctions pour émuler les fonctionnalités d'une base de données

Dans certains cas, les fonctionnalités de base de données ne peuvent pas être converties en fonctionnalités Amazon RDS équivalentes. Par exemple, Oracle envoie des appels par e-mail qui utilisent UTL\_SMTP et Microsoft SQL Server peut utiliser un planificateur de tâches. Si vous hébergez et gérez vous-même une base de données sur Amazon EC2, vous pouvez émuler ces fonctionnalités en les remplaçant par des AWS services.

L'assistant du pack d' AWS SCT extension vous aide à installer, créer et configurer des fonctions Lambda pour émuler le courrier électronique, la planification des tâches et d'autres fonctionnalités.

### Appliquer le pack d'extension pour prendre en charge les fonctions Lambda

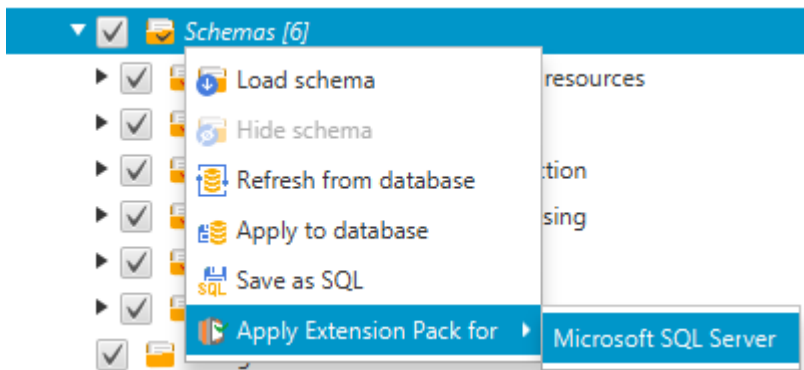
Vous pouvez appliquer le pack d'extension pour prendre en charge les fonctions Lambda à l'aide de l'assistant du pack d'extension ou lorsque vous appliquez le code converti à votre base de données cible.

#### Important

Les fonctionnalités d'émulation de AWS service ne sont prises en charge que pour les bases de données installées et autogérées sur Amazon. EC2 N'installez pas les fonctionnalités d'émulation de service si votre base de données cible se trouve sur une instance de base de données Amazon RDS.

Pour appliquer le pack d'extension à l'aide de l'assistant du pack d'extension

1. Dans l' AWS Schema Conversion Tool arborescence de la base de données cible, ouvrez le menu contextuel (clic droit), choisissez Appliquer le pack d'extension pour, puis choisissez votre plate-forme de base de données source.



L'assistant Kit d'extension s'affiche.

2. Lisez la page d'accueil, puis choisissez Next.
3. Sur la page des paramètres du AWS profil, procédez comme suit :
  - Si vous réinstallez uniquement le schéma du kit d'extension, choisissez Skip this step for now, puis Next.
  - Si vous installez des AWS services, fournissez les informations d'identification pour vous connecter à votre Compte AWS. Vous pouvez utiliser vos AWS CLI informations d'identification si vous les avez AWS CLI installées. Vous pouvez également utiliser les informations d'identification précédemment stockées dans un profil dans les paramètres globaux de l'application et associées au projet. Le cas échéant, choisissez Navigate to Project Settings pour associer un autre profil au projet. Le cas échéant, choisissez Global Settings pour créer un profil. Pour de plus amples informations, veuillez consulter [Gestion des profils dans le AWS Schema Conversion Tool](#).
4. Dans la page Email Sending Service, procédez comme suit :
  - Si vous réinstallez uniquement le schéma du kit d'extension, choisissez Skip this step for now, puis Next.
  - Si vous installez des AWS services et que vous possédez déjà une fonction Lambda, vous pouvez la fournir. Sinon, l'assistant la crée à votre place. Lorsque vous avez terminé, cliquez sur Next.
5. Dans la page Job Emulation Service, procédez comme suit :
  - Si vous réinstallez uniquement le schéma du kit d'extension, choisissez Skip this step for now, puis Next.



- Si vous installez des AWS services et que vous possédez déjà une fonction Lambda, vous pouvez la fournir. Sinon, l'assistant la crée à votre place. Lorsque vous avez terminé, cliquez sur Next.
6. Sur la page d'émulation de fonctions, choisissez Create extension pack. Des messages affichent l'état des opérations du kit d'extension.

Lorsque vous avez terminé, choisissez Finish.

#### Note

Pour mettre à jour un pack d'extension et remplacer les anciens composants du pack d'extension, assurez-vous d'utiliser la dernière version de AWS SCT. Pour de plus amples informations, veuillez consulter [Installation et configuration AWS Schema Conversion Tool](#).

## Configuration des fonctions du pack d' AWS SCT extension

Le pack d'extension contient des fonctions que vous devez configurer avant de les utiliser. La constante `CONVERSION_LANG` définit la langue utilisée par le service pack. Les fonctions sont disponibles en anglais et en allemand.

Pour définir la langue en anglais ou en allemand, apportez la modification suivante au code de fonction. Trouvez la déclaration constante suivante :

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

Pour passer `CONVERSION_LANG` à l'anglais, modifiez la ligne comme suit :

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Pour passer `CONVERSION_LANG` à l'allemand, modifiez la ligne comme suit :

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Définissez ce paramètre pour les fonctions suivantes :

- `aws_sqlserver_ext.conv_datetime_to_string`

- `aws_sqlserver_ext.conv_date_to_string`
- `aws_sqlserver_ext.conv_string_to_date`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.parse_to_date`
- `aws_sqlserver_ext.parse_to_datetime`
- `aws_sqlserver_ext.parse_to_time`

# Bonnes pratiques d'utilisation AWS Schema Conversion Tool

Trouvez des informations sur les meilleures pratiques et les options d'utilisation du AWS Schema Conversion Tool (AWS SCT).

## Configuration de mémoire supplémentaire

Pour convertir des schémas de base de données volumineux, tels qu'une base de données contenant 3 500 procédures stockées, vous pouvez configurer la quantité de mémoire disponible pour le. AWS Schema Conversion Tool

Pour modifier la quantité de mémoire AWS SCT consommée

1. Dans le menu Paramètres, choisissez Paramètres globaux, puis sélectionnez Options JVM.
2. Choisissez Modifier le fichier de configuration et choisissez l'éditeur de texte pour ouvrir le fichier de configuration.
3. Modifiez la section `JavaOptions` afin de définir la quantité de mémoire minimale et maximale disponible. L'exemple suivant définit le minimum à 4 Go et le maximum à 40 Go.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Nous vous recommandons de définir la mémoire minimale disponible à au moins quatre Go.

4. Enregistrez le fichier de configuration, cliquez sur OK, puis redémarrez AWS SCT pour appliquer les modifications.

## Configuration du dossier de projet par défaut

AWS SCT utilise le dossier du projet pour stocker les fichiers du projet, enregistrer les rapports d'évaluation et stocker le code converti. Par défaut, AWS SCT stocke tous les fichiers dans le dossier de l'application. Vous pouvez définir un autre dossier comme dossier de projet par défaut.

## Pour modifier le dossier de projet par défaut

1. Dans le menu Paramètres, choisissez Paramètres globaux, puis chemin du fichier.
2. Dans le champ Chemin du fichier de projet par défaut, entrez le chemin du dossier de projet par défaut.
3. Choisissez Appliquer, puis OK.

## Augmenter la vitesse de migration des données

Pour migrer de grands ensembles de données, tels qu'un ensemble de tables contenant plus de 1 To de données, vous souhaitez peut-être augmenter la vitesse de migration. Lorsque vous utilisez des agents d'extraction de données, la vitesse des migrations de données dépend de différents facteurs. Ces facteurs incluent le nombre de tranches dans votre cluster Amazon Redshift cible, la taille d'un fragment de fichier dans votre tâche de migration, la RAM disponible sur le PC sur lequel vous exécutez vos agents d'extraction de données, etc.

Pour accélérer la migration des données, nous vous recommandons d'exécuter plusieurs sessions de test de migration avec de petits ensembles de données de vos données de production. Nous vous recommandons également d'exécuter vos agents d'extraction de données sur un PC doté d'un SSD d'au moins 500 Go. Au cours de ces sessions de test, modifiez les différents paramètres de migration et surveillez l'utilisation de votre disque afin de déterminer la configuration qui garantit la vitesse maximale de migration des données. Utilisez ensuite cette configuration pour migrer l'ensemble de vos données.

## Augmenter les informations de journalisation

Vous pouvez augmenter les informations de journalisation produites AWS SCT lors de la conversion de vos bases de données, de vos scripts et du code SQL de votre application. Bien que l'augmentation des informations de journalisation puisse ralentir la conversion, les modifications peuvent vous aider à fournir des informations fiables au AWS Support en cas d'erreur.

AWS SCT stocke les journaux dans votre environnement local. Vous pouvez consulter ces fichiers journaux et les partager avec le AWS Support ou les AWS SCT développeurs à des fins de résolution des problèmes.

### Pour modifier les paramètres de journalisation

1. Dans le menu Paramètres, choisissez Paramètres globaux, puis Journalisation.

2. Dans le champ Chemin du dossier des journaux, entrez le dossier dans lequel les journaux seront stockés depuis l'interface utilisateur.
3. Pour le chemin du dossier des journaux de la console, entrez le dossier dans lequel sont stockés les journaux de l'interface de ligne de commande (CLI) d'AWS SCT.
4. Pour Taille maximale du fichier journal (Mo), entrez la taille, en Mo, d'un seul fichier journal. Une fois que votre fichier atteint cette limite, il AWS SCT crée un nouveau fichier journal.
5. Pour Nombre maximal de fichiers journaux, entrez le nombre de fichiers journaux à stocker. Lorsque le nombre de fichiers journaux du dossier atteint cette limite, le fichier journal le plus ancien est supprimé.
6. Pour le chemin de téléchargement du journal des extracteurs, entrez le dossier dans lequel sont stockés les journaux des agents d'extraction de données.
7. Pour le chemin du journal de l'extracteur Cassandra, entrez le dossier dans lequel sont stockés les journaux des agents d'extraction de données Apache Cassandra.
8. Sélectionnez Demander un chemin avant le chargement pour vous assurer qu'AWS SCT demande où stocker les journaux chaque fois que vous utilisez des agents d'extraction de données.
9. Pour le mode Debug, choisissez True. Utilisez cette option pour enregistrer des informations supplémentaires lorsque AWS SCT les journaux standard ne signalent aucun problème.
10. Choisissez les principaux modules d'application pour augmenter les informations de journalisation. Vous pouvez augmenter les informations de journalisation pour les modules d'application suivants :
  - Général
  - Chargeur
  - Analyseur
  - Imprimante
  - Résolveur
  - Télémétrie
  - Convertisseur
  - Cartographie des types
  - Interface utilisateur
  - Contrôleur
  - Comparer le schéma

- Cloner un centre de données
- Analyseur d'applications

Pour chacun des modules d'application précédents, choisissez l'un des niveaux de journalisation suivants :

- Trace : informations les plus détaillées.
- Debug — Informations détaillées sur le flux dans le système.
- Info — Événements d'exécution, tels que le démarrage ou l'arrêt.
- Avertissement — Utilisation d'une API obsolète APIs, mauvaise utilisation de l'API, autres situations d'exécution indésirables ou inattendues.
- Erreur : erreurs d'exécution ou conditions inattendues.
- Critique : erreurs entraînant la fermeture de l'application.
- Obligatoire — Le niveau d'erreur le plus élevé possible.

Par défaut, après avoir activé le mode Debug, AWS SCT définit le niveau de journalisation des informations pour tous les modules de l'application.

Par exemple, pour résoudre les principaux problèmes lors de la conversion, définissez l'analyseur, le mappage des types et l'interface utilisateur sur Trace.

Si les informations deviennent trop détaillées pour le système de fichiers dans lequel les journaux sont diffusés, choisissez un emplacement disposant d'un espace suffisant pour capturer les journaux.

Pour transmettre les journaux au AWS Support, rendez-vous dans le répertoire où ils sont stockés et compressez tous les fichiers dans un seul fichier .zip gérable. Téléchargez ensuite le fichier .zip avec votre dossier d'assistance. Lorsque l'analyse initiale est terminée et que le développement en cours reprend, remettez le mode Debug sur false pour éliminer la journalisation détaillée. Augmentez ensuite la vitesse de conversion.

#### Tip

Pour gérer la taille des journaux et rationaliser les problèmes liés aux rapports, supprimez les journaux ou déplacez-les vers un autre emplacement après une conversion réussie. Cette

tâche garantit que seules les erreurs et informations pertinentes sont transmises au AWS Support et empêche le système de fichiers journaux de se remplir.

# Résolution des problèmes liés à AWS Schema Conversion Tool

Vous trouverez ci-dessous des informations sur la résolution des problèmes liés au AWS Schema Conversion Tool (AWS SCT).

## Je ne peux pas charger d'objets à partir d'une base de données source Oracle

Lorsque vous tentez de charger le schéma à partir d'une base de données Oracle, l'une des erreurs suivantes peut survenir.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Ces erreurs se produisent parce que l'utilisateur dont vous avez utilisé l'identifiant pour vous connecter à la base de données Oracle ne dispose pas des autorisations suffisantes pour lire le schéma, comme l'exige AWS SCT.

Vous pouvez résoudre le problème en accordant à l'utilisateur l'autorisation `select_catalog_role`, ainsi que une autorisation pour n'importe quel dictionnaire dans la base de données. Ces autorisations permettent l'accès en lecture seule pour les vues et les tables système qui est requis par AWS SCT. L'exemple suivant crée un ID d'utilisateur nommé `min_privs` et accorde à l'utilisateur avec cet ID les autorisations minimales requises pour convertir le schéma à partir d'une base de données source Oracle.

```
create user min_privs identified by min_privs;  
grant connect to min_privs;  
grant select_catalog_role to min_privs;  
grant select any dictionary to min_privs;
```

## Message d'avertissement du rapport d'évaluation

Pour évaluer la complexité de la conversion vers un autre moteur de base de données, AWS SCT vous devez accéder aux objets de votre base de données source. Lorsque vous AWS SCT



rencontrez des problèmes lors de l'analyse et que vous ne pouvez pas effectuer d'évaluation, un message d'avertissement est émis. Ce message indique que le pourcentage de conversion global est réduit. Les raisons pour lesquelles des problèmes AWS SCT peuvent survenir lors de la numérisation sont les suivantes :

- L'utilisateur de votre base de données n'a pas accès à tous les objets nécessaires. Pour plus d'informations sur les autorisations et privilèges de sécurité AWS SCT requis pour votre base de données, consultez [Connexion aux bases de données sources à l'aide du AWS Schema Conversion Tool](#) la section de ce guide consacrée à la base de données source appropriée.
- Un objet cité dans le schéma n'existe plus dans la base de données. Pour résoudre le problème, vous pouvez vous connecter avec les autorisations SYSDBA et vérifier si l'objet est présent dans la base de données.
- Le SCT essaie d'évaluer un objet chiffré.

# Référence CLI pour AWS Schema Conversion Tool

Cette section décrit comment démarrer avec l'interface de ligne de commande (CLI) de AWS SCT. Cette section fournit également des informations sur les raccourcis clavier et les modes d'utilisation. Pour une référence complète des commandes AWS SCT CLI, voir [Matériel de référence](#).

## Rubriques

- [Conditions préalables à l'utilisation de l'interface de ligne de commande AWS SCT](#)
- [AWS SCT Mode interactif CLI](#)
- [Obtenir des scénarios AWS SCT CLI](#)
- [Modification de scénarios AWS SCT CLI](#)
- [AWS SCT Mode script CLI](#)
- [AWS SCT Matériel de référence CLI](#)

## Conditions préalables à l'utilisation de l'interface de ligne de commande AWS SCT

Téléchargez et installez la dernière version d'Amazon Corretto 11. Pour plus d'informations, consultez la section [Téléchargements pour Amazon Corretto 11 dans le guide de l'utilisateur d'Amazon Corretto 11](#).

Téléchargez et installez la dernière version de AWS SCT. Pour de plus amples informations, veuillez consulter [Installation AWS Schema Conversion Tool](#).

## AWS SCT Mode interactif CLI

Vous pouvez utiliser l'interface de ligne de commande de AWS SCT en mode interactif. Dans ce mode, vous entrez les commandes une par une dans la console. Vous pouvez utiliser ce mode interactif pour en savoir plus sur les commandes de la CLI ou pour télécharger les scénarios de CLI les plus couramment utilisés.

Pour convertir le schéma de votre base de données source AWS SCT, exécutez une opération séquentielle : créez un nouveau projet, connectez-vous aux bases de données source et cible, créez des règles de mappage et convertissez des objets de base de données. Ce flux de travail pouvant

être complexe, nous vous recommandons d'utiliser des scripts en mode AWS SCT CLI. Pour de plus amples informations, veuillez consulter [Mode script](#).

Vous pouvez exécuter les commandes AWS SCT CLI depuis le app dossier de votre chemin AWS SCT d'installation. Sous Windows, le chemin d'installation par défaut est C : \Program Files\AWS Schema Conversion Tool\. Assurez-vous que ce dossier inclut le AWSSchemaConversionToolBatch.jar fichier.

Pour accéder au mode interactif de la AWS SCT CLI, utilisez la commande suivante après avoir rempli les conditions requises.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Vous pouvez maintenant exécuter des commandes AWS SCT CLI. Assurez-vous de terminer vos commandes par / une nouvelle ligne. Assurez-vous également d'utiliser des guillemets simples ( ' ) avant et après les valeurs des paramètres de commande.

#### Note

Si la commande précédente revient Unexpected error, essayez ce qui suit :

```
java -Djdk.jar.maxSignatureFileSize=200000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Pour voir la liste des commandes disponibles en mode interactif AWS SCT CLI, exécutez la commande suivante.

```
help  
/
```

Pour afficher les informations relatives à une commande AWS SCT CLI, utilisez la commande suivante.

```
help -command: 'command_name'  
/
```

Dans l'exemple précédent, remplacez *command\_name* par le nom d'une commande.

Pour afficher les informations relatives aux paramètres d'une commande AWS SCT CLI, utilisez la commande suivante.

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```

Dans l'exemple précédent, remplacez *command\_name* par le nom d'une commande. Remplacez-le ensuite *parameters\_list* par une liste de noms de paramètres séparés par une virgule.

Pour exécuter un script à partir d'un fichier en mode interactif AWS SCT CLI, utilisez la commande suivante.

```
ExecuteFile -file: 'file_path'  
/
```

Dans l'exemple précédent, remplacez *file\_path* par le chemin d'accès à votre fichier par un script. Assurez-vous que votre fichier possède une `.scts` extension.

Pour quitter le mode interactif de la AWS SCT CLI, exécutez la `quit` commande.

## Exemples

L'exemple suivant affiche les informations relatives à la `Convert` commande.

```
help -command: 'Convert'  
/
```

L'exemple suivant affiche les informations relatives à deux paramètres de la `Convert` commande.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

## Obtenir des scénarios AWS SCT CLI

Pour obtenir les AWS SCT scénarios les plus couramment utilisés, vous pouvez utiliser la `GetCliScenario` commande. Vous pouvez exécuter cette commande en mode interactif, puis modifier les modèles téléchargés. Utilisez les fichiers édités en mode script.

La `GetCliScenario` commande enregistre le modèle sélectionné ou tous les modèles disponibles dans le répertoire spécifié. Le modèle contient l'ensemble complet des commandes permettant

d'exécuter un script. Assurez-vous de modifier les chemins de fichiers, les informations d'identification de base de données, les noms des objets et les autres données de ces modèles. Assurez-vous également de supprimer les commandes que vous n'utilisez pas et d'ajouter de nouvelles commandes au script si nécessaire.

Pour exécuter la `GetCliScenario` commande, remplissez les conditions requises et passez en mode interactif AWS SCT CLI. Pour de plus amples informations, veuillez consulter [Mode interactif](#).

Utilisez ensuite la syntaxe suivante pour exécuter la `GetCliScenario` commande et obtenir les AWS SCT scénarios.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'  
/
```

Dans l'exemple précédent, remplacez-le *template\_type* par l'un des types de modèles du tableau suivant. Remplacez ensuite *file\_path* par le chemin le dossier dans lequel vous souhaitez télécharger les scripts. Assurez-vous qu'il AWS SCT peut accéder à ce dossier sans demander de droits d'administrateur. Assurez-vous également d'utiliser des guillemets simples ( ' ) avant et après les valeurs des paramètres de commande.

Pour télécharger tous les modèles de AWS SCT CLI, exécutez la commande précédente sans l'-typeoption.

Le tableau suivant indique les types de modèles de AWS SCT CLI que vous pouvez télécharger. Pour chaque modèle, le tableau inclut le nom du fichier et la description des opérations que vous pouvez exécuter à l'aide du script.

Type de modèle	Nom de fichier	Description
BTEQScriptConversion	BTEQScriptConversionTemplate.scts	Convertit les requêtes Teradata Basic (BTEQ), FastExport, FastLoad et les scripts Teradata Query en MultiLoad Amazon Redshift RSQL. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de données à l'aide d'ETL</a> .

Type de modèle	Nom de fichier	Description
ConversionApply	ConversionTemplate.scts	Convertit les schémas de base de données source et applique le code converti à la base de données cible. Le cas échéant, enregistre le code converti sous forme de script SQL et enregistre le rapport d'évaluation. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas</a> .
GenericAppConversion	GenericApplicationConversionTemplate.scts	Convertit le code SQL intégré à vos applications à l'aide du convertisseur AWS SCT d'applications générique. Pour de plus amples informations, veuillez consulter <a href="#">Code SQL</a> .
HadoopMigration	HadoopMigrationTemplate.scts	Migre votre cluster Hadoop sur site vers Amazon EMR. Pour de plus amples informations, veuillez consulter <a href="#">Connexion aux bases de données Apache Hadoop à l'aide du AWS Schema Conversion Tool</a> .

Type de modèle	Nom de fichier	Description
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Reprend une migration interrompue de votre cluster Hadoop sur site vers Amazon EMR. Pour de plus amples informations, veuillez consulter <a href="#">Connexion aux bases de données Apache Hadoop à l'aide du AWS Schema Conversion Tool</a> .
Informatica	InformaticaConversionTemplate.scts	Convertit le code SQL intégré dans vos scripts d'extraction, de transformation et de chargement (ETL) Informatica. Configure les connexions à vos bases de données source et cible dans vos scripts ETL, et enregistre les scripts convertis après la conversion. Pour de plus amples informations, veuillez consulter <a href="#">Scripts ETL Informatica</a> .
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Convertit le code SQL intégré dans vos applications C#, C++, Java et Pro*C à l'aide du AWS SCT convertisseur d'applications. Pour de plus amples informations, veuillez consulter <a href="#">Conversion du code SQL d'une application</a> .

Type de modèle	Nom de fichier	Description
OozieConversion	OozieConversionTemplate.scts	Convertit vos flux de travail Apache Oozie en. AWS Step Functions Pour de plus amples informations, veuillez consulter <a href="#">Connexion aux flux de travail Apache Oozie à l'aide du AWS Schema Conversion Tool</a> .
RedshiftAgent	DWHDataMigrationTemplate.scts	Convertit les schémas d'entrepôt de données source et applique le code converti à la base de données Amazon Redshift cible. Enregistre ensuite un agent d'extraction de données, crée et lance une tâche de migration de données. Pour de plus amples informations, veuillez consulter <a href="#">Migration depuis un entrepôt de données</a> .
ReportCreation	ReportCreationTemplate.scts	Crée un rapport de migration de base de données pour plusieurs schémas de base de données source. Enregistre ensuite ce rapport sous forme de fichier CSV ou PDF. Pour de plus amples informations, veuillez consulter <a href="#">Rapport d'évaluation</a> .



Type de modèle	Nom de fichier	Description
SQLScriptConversion	SQLScriptConversionTemplate.scts	Convertit les scripts SQL*Plus ou TSQL en PL/SQL et enregistre les scripts convertis. Enregistre également un rapport d'évaluation.

Après avoir téléchargé le modèle de AWS SCT CLI, utilisez l'éditeur de texte pour configurer le script afin qu'il s'exécute sur vos bases de données source et cible. Utilisez ensuite le mode script AWS SCT CLI pour exécuter votre script. Pour de plus amples informations, veuillez consulter [AWS SCT Mode script CLI](#).

## Exemples

L'exemple suivant télécharge tous les modèles dans le C:\SCT\Templates dossier.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

L'exemple suivant télécharge le modèle de l'ConversionApplyopération dans le C:\SCT\Templates dossier.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

## Modification de scénarios AWS SCT CLI

Après avoir téléchargé les modèles de scénario, configurez-les pour obtenir des scripts fonctionnels pouvant être exécutés sur vos bases de données.

Pour tous les modèles, assurez-vous de fournir le chemin d'accès aux pilotes de vos bases de données source et cible. Pour de plus amples informations, veuillez consulter [Installation des pilotes JDBC pour AWS Schema Conversion Tool](#).

Assurez-vous d'inclure les informations d'identification de base de données pour les bases de données source et cible. Assurez-vous également de configurer des règles de mappage pour décrire

une paire source-cible pour votre projet de conversion. Pour de plus amples informations, veuillez consulter [Cartographie des types de données](#).

Configurez ensuite l'étendue des opérations à exécuter. Vous pouvez supprimer les commandes que vous n'utilisez pas ou ajouter de nouvelles commandes au script.

Supposons, par exemple, que vous envisagiez de convertir tous les schémas de votre base de données Oracle source en PostgreSQL. Vous prévoyez ensuite d'enregistrer le rapport d'évaluation de la migration de votre base de données au format PDF et d'appliquer le code converti à la base de données cible. Dans ce cas, vous pouvez utiliser le modèle pour l'**ConversionApply**opération. Utilisez la procédure suivante pour modifier votre modèle de AWS SCT CLI.

Pour modifier le modèle de AWS SCT CLI pour l'**ConversionApply**opération

1. Ouvrez le fichier `ConversionTemplate.scts` que vous avez téléchargé. Pour de plus amples informations, veuillez consulter [Exemples](#).
2. Supprimez `CreateFilter`, convertissez les opérations `ApplyToTarget -filter`, `-filter`, `SaveTargetSQL`, `SaveTargetSQLbyStatement` et `SaveReportCSV` de votre script.
3. Pour `oracle_driver_file` dans l'`SetGlobalSettings`opération, entrez le chemin d'accès à votre pilote Oracle. Ensuite, pour `postgresql_driver_file`, entrez le chemin d'accès à votre pilote PostgreSQL.

Si vous utilisez d'autres moteurs de base de données, utilisez des noms appropriés pour les paramètres. Pour obtenir la liste complète des paramètres globaux que vous pouvez définir dans le cadre de l'`SetGlobalSettings`opération, voir la matrice des paramètres globaux dans le [Matériel de référence](#).

4. (Facultatif) Pour `CreateProject`, entrez le nom de votre projet et l'emplacement de votre fichier de projet local. Si vous choisissez de continuer avec les valeurs par défaut, assurez-vous que vous AWS SCT pouvez créer des fichiers dans le `C:\temp` dossier sans demander de droits d'administrateur.
5. Pour `AddSource`, entrez l'adresse IP de votre serveur de base de données source. Entrez également le nom d'utilisateur, le mot de passe et le port pour vous connecter à votre serveur de base de données source.
6. Pour `AddTarget`, entrez l'adresse IP de votre serveur de base de données cible. Entrez également le nom d'utilisateur, le mot de passe et le port pour vous connecter à votre serveur de base de données cible.
7. (Facultatif) Pour `AddServerMapping`, entrez les objets de base de données source et cible que vous souhaitez ajouter à une règle de mappage. Vous pouvez utiliser des `targetTreePath`

paramètres `sourceTreePath` et pour spécifier le chemin d'accès aux objets de base de données. Vous pouvez éventuellement utiliser `sourceNamePath` et `targetNamePath` pour spécifier les noms des objets de base de données. Pour plus d'informations, consultez la section [Commandes de mappage de serveurs](#) dans le [Matériel de référence](#).

Les valeurs par défaut de l'AddServerMappingopération mappent tous les schémas source avec votre base de données cible.

8. Enregistrez le fichier, puis utilisez le mode script pour l'exécuter. Pour de plus amples informations, veuillez consulter [Mode script](#).

## AWS SCT Mode script CLI

Après avoir créé un script AWS SCT CLI ou modifié un modèle, vous pouvez l'exécuter à l'aide de la `RunSCTBatch` commande. Assurez-vous d'enregistrer votre fichier avec le script CLI en tant qu'.`sct`extension.

Vous pouvez exécuter des scripts AWS SCT CLI depuis le app dossier de votre chemin AWS SCT d'installation. Sous Windows, le chemin d'installation par défaut est `C:\Program Files\AWS Schema Conversion Tool\`. Assurez-vous que ce dossier inclut le `RunSCTBatch.sh` fichier `RunSCTBatch.cmd` ou. Ce dossier doit également inclure le `AWSSchemaConversionToolBatch.jar` fichier.

Vous pouvez également ajouter le chemin du `RunSCTBatch` fichier dans la variable d'`PATH`environnement de votre système d'exploitation. Après avoir mis à jour la variable d'`PATH`environnement, vous pouvez exécuter des scripts AWS SCT CLI à partir de n'importe quel dossier.

Pour exécuter un script AWS SCT CLI, utilisez la commande suivante sous Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Dans l'exemple précédent, remplacez *file\_path* par le chemin d'accès à votre fichier par un script.

Pour exécuter un script AWS SCT CLI, utilisez la commande suivante sous Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Dans l'exemple précédent, remplacez *file\_path* par le chemin d'accès à votre fichier par un script.

Vous pouvez fournir des paramètres facultatifs dans cette commande, tels que les informations d'identification de la base de données, le niveau de détail dans la sortie de la console, etc. Pour plus d'informations, téléchargez la référence de l'interface de ligne de AWS SCT commande à l'adresse [Matériel de référence](#).

## Exemples

L'exemple suivant exécute le `ConversionTemplate.scts` script dans le `C:\SCT\Templates` dossier. Vous pouvez utiliser cet exemple sous Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

L'exemple suivant exécute le `ConversionTemplate.scts` script dans le `/home/user/SCT/Templates` répertoire. Vous pouvez utiliser cet exemple sous Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

## AWS SCT Matériel de référence CLI

Vous trouverez des informations de référence sur l'interface de ligne de AWS Schema Conversion Tool commande (CLI) dans le guide suivant : [Référence de la AWS Schema Conversion Tool CLI](#).

# Notes de mise à jour pour AWS Schema Conversion Tool

Cette section contient les notes de publication pour AWS SCT, à partir de la version 1.0.640.

## Notes de mise à jour pour la AWS SCT version 676

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
Oracle	PostgreSQL/ Aurora PostgreSQL	Nouvelle émulation de fonction intégrée pour les fonctions suivantes : <ul style="list-style-type: none"> <li>• SYS.UTL_RAW.BIT_AND(RAW, RAW)</li> <li>• XDB.DBMS_XSLPROCESSOR.CLOB2FILE(CLOB)</li> <li>• XDB.DBMS_XSLPROCESSOR.READ2CLOB(VARCHAR2)</li> <li>• SYS.UTL_RAW.BIT_OR(RAW, RAW)</li> <li>• SYS.UTL_RAW.BIT_COMPLEMENT(RAW)</li> </ul>	Non	Oui
Serveur MS SQL	Serveur SQL Amazon RDS	Database Mail not supported Message supprimé du rapport PDF	Oui	Oui
Oracle	PostgreSQL/ Aurora PostgreSQL	Conversion des contraintes implémentée pour les tables partitionnées.	Oui	Oui

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
Oracle	MySQL	Examen de l'applicabilité de l'AI-602 à la conversion de tables	Oui	Oui
Serveur MS SQL	PostgreSQL/Aurora PostgreSQL	prend désormais en charge la MERGE déclaration dans PostgreSQL 15.x	Oui	Oui
Tous	Tous	Connexions JDBC implémentées : propriétés avancées	Oui	Non
Tous	Tous	CLI : échec de <code>PrintOLAPTaskStatus</code> commande corrigé	Oui	Non
Teradat	Amazor	Implémentation du type de données de type Teradata.	Oui	Non
Teradat	Amazor	Correction d'une MERGE conversion incorrecte dans SQL/BTEQ.	Oui	Non
Teradat	Amazor	Implémentation du type de données de type Teradata.	Oui	Non
Teradat	Amazor	Conversions LEAD/LAG de fonctions implémentées.	Oui	Non
Teradat	Amazor	Correction d'une erreur AI-9996 <code>Transformer error occurred in statement .</code>	Oui	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
Teradat	Amazor	Correction d'une erreurAI-9996 Transformer error in selectItem .	Oui	Non
Teradat	Amazor	Conversion implémentée pour la procédure stockée partielle : XbidQM.SpCmprsnDly	Oui	Non
Teradat	Amazor	UNPIVOTDéclaration implémentée avec alias.	Oui	Non
Teradat	Amazor	DeleteDéclaration implémentée avec plusieurs tables sources.	Oui	Non
Teradat	Amazor	Corriger pourAI-9996 Transformer error occurred in functionCallExpression .	Oui	Non
Teradat	Amazor	Conversion de NORMALIZE clauses implémentée.	Oui	Non
Teradat	Amazor	Correction d'une conversion incorrecte dans les DELETE instructions contenant des sous-requêtes.	Oui	Non
Teradat	Amazor	Correction d'une erreurAI-9996 Transformer error occurred in tableOperatorSource .	Oui	Non
Teradat	Amazor	Correction d'une erreurAI-9996 Transformer error occurred in additiveExpression .	Oui	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
Teradat	Amazon	Conversion d'objets du système DBC implémentée.	Oui	Non
Teradat	Amazon	Solution de mise à jour implémentée avec des prédicats de jointure implicites.	Oui	Non
Netezza	Amazon	Correction d'CREATE MATERIALIZED VIEW une erreur de conversion des relevés.	Oui	Non
DB2LUW	PostgreSQL/ Aurora PostgreSQL	Connexion JDBC Extended Options : options de connexion supplémentaires ajoutées.	Oui	Non
DB2LUW	PostgreSQL/ Aurora PostgreSQL	Ajout de la prise en charge de MERGE Statement dans PostgreSQL 15.x	Oui	Non
DB2LUW	PostgreSQL/ Aurora PostgreSQL	GLOBAL TEMPORARY TABLE Conversion mise en œuvre.	Oui	Non



Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
DB2LUW	PostgreSQL/Aurora PostgreSQL	USER DEFINED TYPES Conversion mise en œuvre.	Oui	Non
DB2LUW	MySQL	GLOBAL TEMPORARY TABLE Conversion mise en œuvre.	Oui	Non
DB2LUW	MySQL	USER DEFINED TYPES Conversion mise en œuvre.	Oui	Non
DB2LUW	MySQL	USER DEFINED FUNCTIONS Conversion mise en œuvre.	Oui	Non
DB2LUW	MariaDB	GLOBAL TEMPORARY TABLE Conversion mise en œuvre.	Oui	Non
DB2LUW	MariaDB	USER DEFINED TYPES Conversion mise en œuvre.	Oui	Non
Sybase	Tous	Ajout de la prise en charge de l'authentification Kerberos	Oui	Non
DB2LUW	PostgreSQL/Aurora PostgreSQL	Ajout du support pour la conversion multiversion pour les cibles	Oui	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité dans AWS DMS Schema Conversion Tool (SCT)	Disponibilité lors de la conversion de AWS DMS schéma
Azure SQL/ Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL	Ajout du support pour la conversion multiversion pour les cibles	Oui	Non
DB2LUW	PostgreSQL/Aurora PostgreSQL	Ajout de la prise en charge des MERGE instructions dans PostgreSQL 15.x.	Oui	Non
Teradata	Amazon	Correction d'une conversion de changement de fonction non prise en charge.	Oui	Non
Tous	Amazon	Extracteurs de données : partitionnement implémenté par une colonne indexée.	Oui	Non

## Notes de mise à jour pour la AWS SCT version 675

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Cassandra	DynamoD	Correction d'un bug à cause duquel l'installation de Cassandra échouait sur le centre de données cible.	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
DB2 LUW	PostgreSQL	SQL DYNAMIQUE : instruction PREPARE : résolution et conversion sans SQL dynamique.	Non
DB2 LUW	PostgreSQL	Ajout du support pour SPECIAL REGISTER.	Non
DB2 LUW	PostgreSQL	Mise à jour du pack d'extension	Non
Hadoop	Amazon EMR	Ajout de la prise en charge de la connexion à un cluster Hadoop via le protocole rsa-sha2.	Non
Microsoft SQL Server	Amazon Fargate	Correction du fait que le pilote JDBC forçait le protocole TLS alors qu'il n'était pas configuré.	Non
Netezza	Amazon Fargate	Ajout du support pour la conversion des vues matérialisées.	Non
Oracle	Amazon Fargate	Ajout de la prise en charge des requêtes récursives dans Amazon Redshift.	Oui
Oracle	PostgreSQL, Aurora PostgreSQL	Correction d'une conversion incorrecte du type de données NUMBER.	Oui
Oracle	Amazon Fargate	Migration des données. Partitionnement automatique Oracle. Date d'expiration ajoutée pour la valeur des fragments de table. Le délai d'expiration est de 72 heures. En cas d'expiration, les fragments de données sont reconstruits lors de la création d'une tâche de migration de données.	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Oracle	Amazon F	SCT Data Extractor : modification de l'approche du téléchargement de données vers Amazon Redshift. Par défaut, l'extracteur ne crée pas de tables intermédiaires. Au lieu de cela, une fois que tous les fichiers de données se trouvent dans le compartiment Amazon S3, l'extracteur les copie dans la table cible à l'aide d'une seule commande COPY.	Non
Oracle	Amazon F	Ajout de la migration du type de données RAW vers la colonne VARBYTE.	Non
Oracle	PostgreSQL, Aurora PostgreSQL	Conversion multiversion	Non
Oracle	PostgreSQL	Ajout de la prise en charge de l'instruction MERGE dans PostgreSQL 15.x.	Oui
Oracle	PostgreSQL	Ajout de la prise en charge des nouvelles fonctions d'expression régulière dans PostgreSQL 15.x.	Oui
Oracle	PostgreSQL, Aurora PostgreSQL	L'instruction ON CONFLICT DO UPDATE est convertie sans alias exclu.	Oui
Teradata	Amazon F	Ajout du support de conversion pour les fonctions LEAD/LAG.	Non
Teradata	Amazon F	Diffusion améliorée des types de données avec indication explicite du format des données.	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Teradata	Amazon F	Conversion améliorée de la clause AT 'TIME ZONE' dans les expressions time/timestamp.	Non
Teradata	Amazon F	AI-9996 pendant les procédures de conversion avec des instructions MERGE.	Non

## Notes de mise à jour pour la AWS SCT version 674

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Tous	Tous	Diverses corrections de bogues et améliorations des performances	Partiel (uniquement pour les paires source et cible prises en charge)
Azure SQL/ Microsoft SQL Server	Amazon F	Suppression du message « AI 18066 : Impossible de convertir le nom du schéma » qui induisait les utilisateurs en erreur lors de l'évaluation/de la conversion du schéma	Non
Azure SQL/ Microsoft SQL Server	Amazon RDS pour MySQL/ Amazon Aurora MySQL	Conversion incorrecte de la procédure sans attribution de code de retour	Partiel (la conversion de schéma ne prend actuellement pas en charge Azure SQL en tant que source)

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Azure SQL/ Micro soft SQL Server	Amazon RDS pour PostgreSQL/ Amazon Aurora PostgreSQL	Corrigé AI9997 pour certains cas de conversion de la clause FOR XML PATH	Partiel (la conversion de schéma ne prend actuellement pas en charge Azure SQL en tant que source)
Azure SQL/ Micro soft SQL Server	Amazon RDS pour PostgreSQL/ Amazon Aurora PostgreSQL	La valeur est arrondie à l'échelle d'origine dans le corps de la procédure/fonction	Partiel (la conversion de schéma ne prend actuellement pas en charge Azure SQL en tant que source)
Azure SQL/ Micro soft SQL Server	Amazon RDS pour PostgreSQL/ Amazon Aurora PostgreSQL	Diverses améliorations apportées à la conversion des instructions EXECUTE	Partiel (la conversion de schéma ne prend actuellement pas en charge Azure SQL en tant que source)

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Azure SQL/ Microsoft SQL Server/ Azure Synapse	Amazon F	<p>Conversion améliorée des instructions et modes suivants :</p> <ul style="list-style-type: none"> <li>• BLOC D'EXCEPTION</li> <li>• AUTOCOMMIT</li> <li>• NONATOMIC</li> <li>• ENSEMBLE DE REGROUPEMENT</li> <li>• CUBE</li> <li>• ROLLUP</li> </ul>	Non
DB2 LUW	Amazon RDS pour PostgreSQL/ Amazon Aurora PostgreSQL	Diverses corrections apportées aux requêtes SQL de chargement de métadonnées	Non
DB2 LUW	Amazon RDS pour PostgreSQL/ Amazon Aurora PostgreSQL	L'IA 9996 n'est pas attendue sur les déclencheurs	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
DB2 z/OS	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Fonction analytique ROWNUMBER	Non
DB2 z/OS	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Support des constantes de chaînes hexadécimales	Non
DB2 z/OS	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Diverses corrections apportées aux métadonnées chargent les requêtes SQL.	Non



Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
DB2 z/ OS	Amazon RDS pour PostgreSQL L/ Amazon Aurora PostgreSQL L	VALEUR SUIVANTE POUR le support de référence de séquence	Non
DB2 z/ OS	Amazon RDS pour PostgreSQL L/ Amazon Aurora PostgreSQL L	Prise en charge de l'option DB2 _NUMBER_ROWS de l'instruction GET DIAGNOSTICS	Non
DB2 z/ OS	Amazon RDS pour PostgreSQL L/ Amazon Aurora PostgreSQL L	OBTENIR DES DIAGNOSTICS Plusieurs instructions	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
DB2 z/OS	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Correction de bogues dans la conversion des instructions FOR.	Non
Oracle	Amazon RDS pour MySQL / Amazon Aurora MySQL	Correction d'un bug lorsque le nœud de paramètre de la fonction du package n'est pas défini.	Oui
Oracle	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Correction de bugs dans la fonction AWS_ORACLE_EXT.NEXT_DAY du pack d'extension	Oui

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
Oracle	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Correction de divers bogues liés à la conversion « (+) » dans les jointures externes d'Oracle	Oui
Oracle		Support de l'authentification Kerberos	Non
SAP ASE	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Correction d'un bug lors de la conversion de plusieurs identifiants dans la clause FROM dans l'instruction UPDATE	Non
SAP ASE	Amazon RDS pour PostgreSQL / Amazon Aurora PostgreSQL	Correction d'un bug avec la conversion des commentaires et des déclarations sur plusieurs lignes	Non

Source	Cible	Nouveautés, améliorations ou correctifs	Disponibilité lors de la conversion de AWS DMS schéma
SAP ASE		Ajout du support pour le paramètre ENCRYPT_PASSWORD lors de la connexion	Non
Teradata	Amazon F	Conversion améliorée de la table VOLATILE avec le nom de schéma spécifié	Non
Teradata	Amazon F	Conversion incorrecte de la clause WHERE dans un CTE complexe	Non
Teradata	Amazon F	Ajout de la prise en charge du type de données INTERVAL lors de la migration de données à l'aide des agents d'extraction de données SCT.	Non
Scripts BTEQ Teradata	Scripts RSQL Amazon Redshift	Paramètres de sortie de conversion incorrects dans une procédure exécutée par BTEQ	Non

## Notes de mise à jour pour la AWS SCT version 673

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Corrections de bogues généraux et améliorations des performances
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Correction d'une conversion incorrecte des appels de fonction
Azure SQL/Micro	Aurora PostgreSQL	Conversion implémentée de la clause FOR XML

Source	Cible	Nouveautés, améliorations ou correctifs
soft SQL Server	L/ Amazon RDS PostgreSQL	
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Conversion d'une clause FOR XML avec un alias incorrect.
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Correction d'un bug AWS SCT empêchant la conversion EXECUTE des instructions qui exécutent une chaîne de caractères avec des paramètres de procédure.
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Conversion améliorée des instructions UPDATE avec des jointures internes.
Azure Synap	Amazon Red	Correction d'une conversion incorrecte de la fonction OBJECT_ID intégrée.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	<p>Implémentation de la conversion des instructions et objets suivants :</p> <ul style="list-style-type: none"> <li>• DECLARE TEMPORARY TABLE statement</li> <li>• DROP TABLE statement</li> <li>• PK et UNIQUE contraintes sur les tables partitionnées</li> <li>• Fonction TIMESTAMPDIFF</li> <li>• Fonction TO_DATE</li> <li>• Fonction EBCDIC_STR</li> <li>• Fonction VARCHAR_FORMAT</li> </ul>
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Correction d'un bug lorsque l'index basé sur les fonctions ignore les fonctions après la conversion.
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Correction d'un bug où l'REPEAT instruction se fermait avec AI 9996 après la conversion
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Correction d'un bug où la FINAL TABLE clause se terminait par 9996.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	LOADER  Clé de partitionnement dans la contrainte de références. AWS SCT peut désormais convertir les clés primaires et les contraintes uniques des tables partitionnées en index secondaires.
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Support de la fonction PostgreSQL.VarChar_Format
IBM DB2 pour z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Implémentation du changement de classement dans les commandes ModifyTransformationRule SCT CLI CreateTransformationRule et SCT.
Greenplum	Amazon Redshift	Correction d'un bug avec un appel incorrect de la procédure stockée après la conversion
Hadoop	Amazon EMF	Ajout de la prise en charge de la connexion à un cluster Hadoop à l'aide du protocole rsa-sha2.
Hadoop	Amazon EMF	Ajout du support pour Amazon EMR avec un métastore autre que Glue Hive,
Oracle	Amazon Redshift	Correction d'un bug avec une conversion incorrecte d'une requête récursive où la colonne PRIOR ne figurait pas dans la liste SELECT.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL L	Implémentation du renvoi d'un élément d'un tableau associatif
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL L	Correction d'un AI 9996 inattendu entre UNPIVOT crochets
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL L	Correction d'un AI 9996 inattendu avec UNPIVOT UNION ALL
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL L	améliorations pour les conversions Number de types de données
Oracle	Extracteur de données Amazon Redshift	Support pour le partitionnement automatique des tables Oracle. Optimisation pour la création de tâches de migration.
Teradata	Amazon Red	Implémenter la conversion de la EXCEPTION BLOCK déclaration



Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift	Support pour la conversion de ALLANY, et de SOME prédicats vers Amazon Redshift.
Teradata	Amazon Redshift	Ajout du support natif pour le QUALIFY prédicat.
Teradata	Amazon Redshift	Conversion améliorée des éléments suivants : <ul style="list-style-type: none"> <li>• Requêtes récursives</li> <li>• GROUPING SET</li> <li>• CUBE</li> <li>• ROLLUP</li> <li>• UPDATE déclaration avec jointure implicite</li> </ul>
Sources OLAP	Extracteur de données Amazon Redshift	Commandes CLI implémentées pour les tâches Stop/Resume for Amazon Redshift Data Extractor.
Sources OLAP	Extracteur de données Amazon Redshift	Ajout de la possibilité de sélectionner les colonnes de table qui doivent être migrées lors de la configuration de la tâche de migration.

## Notes de mise à jour pour la AWS SCT version 672

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Amazon RDS for PostgreSQL	Support implémenté de la version majeure 15 de PostgreSQL en tant que cible de migration.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Amazon Red	Ajout d'une nouvelle <code>PrintTaskStatus</code> commande dans l'interface de ligne de commande de AWS SCT (CLI) pour afficher l'état de la tâche de migration des données.
Tous	Amazon Red	Amélioration du flux de configuration pour les agents d'extraction de données.
Tous	Amazon Red	Résolution d'une erreur en raison de laquelle les agents d'extraction de données n'affichaient pas les informations relatives aux sous-tâches.
Apache Oozie	AWS Step Functions	Ajout d'une option permettant d'enregistrer les définitions des machines à états sous forme de script dans le code converti.
Base de données Azure SQL	Aurora PostgreSQL	Implémentation de la conversion des SUM fonctions COALESCE DATEADDGETDATE,, et.
Microsoft SQL Server	PostgreSQL	
Base de données Azure SQL	Aurora PostgreSQL	Amélioration de la conversion des UPDATE instructions comportant des OUTPUT clauses JOIN et.
Microsoft SQL Server	PostgreSQL	
Base de données Azure SQL	Aurora PostgreSQL	Résolution d'une erreur survenue lors de la conversion de l' <code>SELECT TOP 1 WITH TIES</code> instruction.
Microsoft SQL Server	PostgreSQL	

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Azure SQL	Aurora PostgreSQL	Plusieurs problèmes survenus lors de la conversion des FOR XML clauses dans les fonctions intégrées ont été résolus.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Implémentation de la conversion GET DIAGNOSTICS et RAISE des instructions à l'aide d'un bloc Amazon Redshift EXCEPTION natif.
Greenplum	Amazon Redshift	Amélioration de la conversion des procédures stockées en ajoutant la prise en charge d'un EXCEPTION bloc dans le code converti.
IBM DB2 pour z/OS	Aurora PostgreSQL PostgreSQL	Correction d'une erreur à cause de laquelle les TO_CHAR fonctions associées aux modèles de format horaire étaient mal converties.
IBM DB2 pour z/OS	Aurora PostgreSQL PostgreSQL	Implémentation de la conversion des expressions de table imbriquées.
IBM DB2 pour z/OS	Aurora PostgreSQL PostgreSQL	Implémentation de la conversion des SIGNAL instructions GOTO MERGEREPEAT,, et.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion FETCH des déclarations avec BEFORE des mots clés AFTER d'orientation.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des références aux OLD TABLE tables FINAL TABLE et.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	<p>Implémentation de la conversion des fonctions suivantes.</p> <ul style="list-style-type: none"> <li>• ADD_MONTHS</li> <li>• DAY avec les paramètres du type de données du caractère</li> <li>• DAYOFWEEK</li> <li>• DAYS</li> <li>• DECODE</li> <li>• HOUR</li> <li>• LAST_DAY</li> <li>• LOCATE_IN_STRING</li> <li>• MICROSECOND</li> <li>• MINUTE</li> <li>• MONTH</li> <li>• ROUND</li> <li>• TIME</li> <li>• TIMESTAMP</li> <li>• TIMESTAMP_FORMAT</li> <li>• TRANSLATE</li> <li>• UNICODE_STR</li> <li>• XMLCAST</li> <li>• XMLELEMENT</li> <li>• XMLQUERY</li> <li>• XMLSERIALIZE</li> <li>• YEAR</li> </ul>
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion de l'alias d'une sous-requête en JOIN clauses.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des COALESCE fonctions.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des EXPLICIT index.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des noms de colonnes en expressions composées afin de résoudre un problème d'apparition inattendue de l'élément d'action 9997 lors de la conversion.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des clés primaires et des contraintes uniques.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des XMLTABLE instructions en INSERT instructions afin de résoudre un problème où l'action 9996 apparaît de manière inattendue pendant la conversion.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion de fonctions avec l'SUBSTRargument.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Problème résolu : l'action 9996 s'affichait de manière inattendue lors de la conversion du registre CURRENT_TIMESTAMP spécial.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème selon lequel l'action 9996 apparaissait de manière inattendue lors de la conversion d'MERGEinstructions, d'instructions non prises en charge et de fonctions intégrées non prises en charge.
Microsoft SQL Server	Tous	Ajout du support de Microsoft SQL Server version 2022 en tant que source.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des SELECT instructions utilisant des opérateurs de concaténation de chaînes. AWS SCT utilise la STRING_AGG fonction dans le code converti.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL L	Support implémenté de la nouvelle version 3.1.0 du fichier de configuration des fonctionnalités de Babelfish. Ce fichier définit les fonctionnalités SQL qui sont prises en charge et non prises en charge par des versions spécifiques de Babelfish.

Source	Cible	Nouveautés, améliorations ou correctifs
Netezza	Amazon Red	Résolution d'un problème en raison duquel les agents d'extraction de données ne démarraient pas la migration des données à partir du point CDC spécifié.
Oracle	Tous	Mise à jour du rapport d'évaluation pour les bases de données Oracle version 19 en tant que source.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion DBMS_OUTPUT du package en ajoutant de nouvelles fonctions au pack d' AWS SCT extension.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion de fonctions et de procédures utilisant des tableaux associatifs comme arguments ou paramètres.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des DISTINCT clauses dans SELECT les déclarations.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des tables où la contrainte de clé primaire porte le même nom que la table.



Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion de la RAISE_APPLICATION_ERROR procédure avec le troisième paramètre.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème en raison duquel la règle de migration ne modifiait pas automatiquement le type de NUMERIC données selon le INTEGER cas.
Oracle DW	Amazon Redshift	Support implémenté des CONNECT BY clauses Amazon Redshift natives dans le code converti.
Oracle DW	Amazon Redshift	Amélioration de la migration des données en ajoutant automatiquement une sous-tâche pour chaque table ou partition dans le périmètre de migration. Cette approche empêche la perte de données pour les données insérées après le partitionnement.
Teradata	Amazon Redshift	Implémentation de la conversion des vues récursives.
Teradata	Amazon Redshift	Amélioration de la conversion des procédures stockées qui utilisent des modes de ANSI transaction BTET et des modes de transaction en ajoutant la prise en charge du mode de AUTOCOMMIT transaction natif Amazon Redshift.
Teradata	Amazon Redshift	Amélioration de la conversion des procédures stockées qui utilisent la sémantique des TERADATA transactions en ajoutant le NONATOMIC mot-clé dans le code converti.
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel le code converti incluait l'ID de clé d' AWS accès et la clé d'accès secrète.

## Notes de mise à jour pour la AWS SCT version 671

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Correction d'une erreur en raison de laquelle AWS SCT ne n'était pas autorisé à enregistrer un fichier de projet sous Windows.
Tous	Tous	<p>Les modèles d'interface de ligne de commande de AWS SCT (CLI) suivants ont été mis à jour.</p> <ul style="list-style-type: none"> <li>• BTEQScriptConversion</li> <li>• ConversionApply</li> <li>• HadoopMigration</li> <li>• HadoopResumeMigration</li> <li>• Informatica</li> </ul> <p>Pour plus d'informations sur les modèles de AWS SCT CLI, consultez <a href="#">Obtenir des scénarios CLI</a>.</p>
Tous	Amazon Red	Correction d'une erreur qui empêchait AWS SCT de créer un pack d'extension dans l'interface de ligne de commande (CLI).
Tous	Amazon Red	Résolution d'un problème selon lequel les agents d'extraction de données n'utilisaient pas la configuration AWS Snowball Edge dans l'interface de ligne de commande (CLI).
Apache Oozie	AWS Step Functions	Support implémenté pour la migration d'Apache Oozie vers AWS Step Functions le mode interface de ligne de commande (CLI). Après avoir migré vos charges de travail Hadoop vers Amazon EMR, vous pouvez désormais migrer le système de planification des flux de travail vers le. AWS Cloud Pour de plus amples informations, veuillez consulter <a href="#">Conversion des flux de travail Oozie</a> ;.
Base de données Azure SQL	Aurora PostgreSQL	Correction d'une erreur de résolution survenue pour les tables et les alias.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	PostgreSQL	
Base de données Azure SQL	Aurora PostgreSQL	Implémentation de la conversion des INDEX ON clauses.
Microsoft SQL Server	PostgreSQL	
Base de données Azure SQL	Aurora PostgreSQL	Amélioration de la conversion des objets suivants afin d'éviter des actions inattendues. <ul style="list-style-type: none"> <li>• Relevés par lots</li> <li>• Listes d'expressions</li> <li>• Alias de table</li> <li>• Tables temporaires</li> <li>• Déclencheurs</li> <li>• Variables utilisateur</li> </ul>
Base de données Azure SQL	Aurora PostgreSQL	Résolution d'une erreur d'analyse qui s'est produite dans les procédures.
Microsoft SQL Server	PostgreSQL	

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Azure SQL	Aurora PostgreSQL	Correction d'une erreur lors de l' AWS SCT utilisation de noms incorrects de tables temporaires dans le code converti pour les OBJECT_ID fonctions.
Microsoft SQL Server	PostgreSQL	
Base de données Azure SQL	Aurora PostgreSQL	Résolution de problèmes liés à l'apparition inattendue de l'action 9996 lors de la conversion des éléments de code suivants.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> <li>• Fonctions CONVERT</li> <li>• Fonctions DATEADD</li> <li>• DELETE instructions dans les fonctions en ligne</li> <li>• Instructions IF</li> <li>• INSERT ou UPDATE des actions sur une colonne</li> <li>• Instructions RETURN</li> <li>• UPDATE instructions contenant des requêtes ou des fonctions complexes</li> </ul>
BigQuery	Amazon Redshift	Ajout du support BigQuery en tant que source pour le processus d'évaluation multiserveur. Pour de plus amples informations, veuillez consulter <a href="#">Rapport d'évaluation multiserveur</a> .
Hadoop	Amazon EMR	Mise à jour de la version du pilote JDBC Apache Hive pris en charge que vous utilisez pour vous connecter à vos bases de données sources. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a> .

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Le chargeur de métadonnées source a été amélioré pour s'assurer qu'il AWS SCT charge les objets de la base de données source tels que les clés primaires, les index implicites, etc.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur de résolution qui se produisait pour les colonnes des curseurs implicites.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la possibilité de conserver le formatage des noms de colonnes, des expressions et des clauses dans les instructions DML du code converti.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des clés étrangères entre schémas.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion LENGTH et VARCHAR des fonctions.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des DECLARE CONDITION instructions LABEL ON et.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion SELECT des instructions avec OPTIMIZE FOR des clauses.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des CREATE TABLE instructions en ajoutant des valeurs par défaut pour tous les types de données pris en charge.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des INCREMENT BY attributs.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des tables partitionnées en ajoutant la possibilité d'exclure les partitions de table du champ de conversion.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des définitions de clés primaires avec des INCLUDE colonnes.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion de la SUBSTRING fonction.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des DECLARE HANDLER FOR relevés SET et des relevés.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des types de données variables.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des XMLTABLE fonctions.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration du flux de migration en implémentant l'ordre suivant pour appliquer les objets convertis à la base de données cible : tables, partitions, index, contraintes, clés étrangères et déclencheurs.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion des commentaires dans le code source.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9997 lors de la conversion d'alias en FROM clauses.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9997 lors de la conversion des alias de curseur.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur à cause de laquelle le code converti renvoyait des résultats différents pour les SELECT instructions contenant ORDER BY des clauses. SQL Server et PostgreSQL NULL traitant les valeurs différemment, le code converti NULLS FIRST inclut désormais des clauses NULLS LAST ou des clauses garantissant que le code converti renvoie les résultats dans le même ordre que le code source.



Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des types de données dans les fonctions de table.
MySQL	Amazon RDS for MySQL	Résolution d'un problème en raison duquel des guillemets simples ( ' ') apparaissaient de manière inattendue autour des noms d'objets de base de données dans le code converti.
Oracle	Aurora PostgreSQL L  PostgreSQL L	De nouvelles vues ont été ajoutées au pack d'extension pour émuler les vues du système Oracle que vous utilisez pour afficher des informations sur les partitions et les sous-partitions.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Mise à jour de deux fonctions du pack d'extension pour ajouter des noms de schéma en tant qu'arguments dans le code converti.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur qui AWS SCT empêchait d'utiliser les paramètres corrects pour la conversion des applications C++ après avoir actualisé le code de l'application dans l'interface utilisateur.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des CREATE TYPE instructions pour éviter les exceptions inattendues.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des tables imbriquées.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'une erreur d'analyse qui s'est produite pour les objets du package.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de découpe AWS SCT inattendue des noms d'objets dans le code converti lorsque la longueur du nom dépasse 60 caractères.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des déclencheurs au niveau des lignes pour les tables partitionnées.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle DW	Amazon Red	Support implémenté du partitionnement automatique des tables pour la migration des données. Pour accélérer la migration des données, AWS SCT vous pouvez partitionner automatiquement de grandes tables ou partitions en fonction des valeurs de la ROWID pseudocolonne. Pour de plus amples informations, veuillez consulter <a href="#">Utilisation du partitionnement natif</a> .
Teradata	Amazon Red	Implémentation de la prise en charge des MERGE commandes natives dans le code Amazon Redshift converti. Pour plus d'informations sur la MERGE commande dans Amazon Redshift, consultez <a href="#">MERGE</a> dans le manuel Amazon Redshift Database Developer Guide.
Teradata	Amazon Red	Amélioration de la conversion des UPDATE instructions DELETE et des instructions qui n'utilisent pas de noms de table explicites.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte NOT IN des instructions IN et instructions.

## Notes de mise à jour pour la AWS SCT version 670

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Azure SQL	Aurora PostgreSQL	Résolution de problèmes liés à l'apparition inattendue de l'action 9996 lors de la conversion des éléments de code suivants.
Microsoft SQL Server	L	<ul style="list-style-type: none"> <li>• CREATE INDEX déclarations contenues dans INCLUDE des déclarations</li> <li>• Instructions DECLARE</li> <li>• Instructions DECLARE . . . TABLE</li> <li>• DECLARE avec des valeurs par défaut dans LOOP les instructions</li> <li>• Instructions DELETE</li> <li>• DROP CONSTRAINT déclarations contenues dans ALTER TABLE des déclarations</li> </ul>

Source	Cible	Nouveautés, améliorations ou correctifs
		<ul style="list-style-type: none"> <li>• EXECUTE AS CALLER et REVERT</li> <li>• Instructions IIF</li> <li>• Listes d'expressions</li> <li>• Fonctions MONTH( )</li> <li>• Instructions UPDATE</li> <li>• Fonctions YEAR( )</li> </ul>
Analyses Azure Synapse	Amazon Red	Ajout de la prise en charge d'Azure Synapse Analytics en tant que source pour le processus d'évaluation multiserveur. Pour de plus amples informations, veuillez consulter <a href="#">Rapport d'évaluation multiserveur</a> .
Hadoop	Amazon EMF	Support implémenté pour la migration des clusters Hadoop vers Amazon EMR en mode interface de ligne de commande (CLI). Pour de plus amples informations, veuillez consulter <a href="#">Migration des infrastructures de mégadonnées</a> .
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur de résolution qui s'est produite pour les tables et les colonnes sources.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des CASE expressions.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Mise en œuvre de la conversion des CURRENT_DATE références en registres spéciaux. Une référence à un registre spécial dans Db2 pour z/OS est une référence à une valeur fournie par le serveur actuel.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion DATE et POSSTR des fonctions.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des constantes date/heure.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des valeurs par défaut pour les colonnes des types de données suivants : DATETIME, TIMESTAMP , etTIMESTAMP WITH TIME ZONE.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion des SELECT INTO instructions.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des DATEDIFF fonctions.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur lors de la conversion des ISNULL fonctions enNULLIF. Par conséquent, le code converti a produit des résultats différents par rapport au code source. Maintenant, AWS SCT convertit ISNULL les fonctions enCOALESCE.
Netezza	Amazon Red	Agents d'extraction de données améliorés pour résoudre un problème selon lequel le statut d'échec était défini pour les tâches achevées avec succès.
Netezza	Amazon Red	Ajout de la possibilité de modifier les points de terminaison dans les sous-tâches après le démarrage d'une migration de données à l'aide d'agents d'extraction de données.
Microsoft SQL Server  MySQL  Oracle  PostgreSQL L	Aurora MySQL  Aurora PostgreSQL L  MySQL  PostgreSQL L	Ajout de la possibilité de se connecter aux bases de données à l'aide d'un protocole d' IPv6 adresse.
Oracle	Amazon RDS for Oracle	Implémentation de la conversion DBMS_JOB du package qui planifie et gère les tâches dans la file d'attente des tâches.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	De nouvelles fonctions ont été ajoutées au pack d'extension pour améliorer la conversion des tables imbriquées globales. Ces nouvelles fonctions DELETE émulent EXTEND et fonctionnent TRIM dans votre code source Oracle.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Ajout de la possibilité de spécifier l'étendue de conversion pour le code SQL intégré dans les applications Java. Vous pouvez désormais exclure les sous-ensembles du projet d'application source de la zone de conversion. Pour de plus amples informations, veuillez consulter <a href="#">Conversion du code SQL de votre application Java en AWS SCT</a> .
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des opérateurs de concaténation (  ) dans les index fonctionnels.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des IN conditions dans lesquelles votre code source n'inclut pas de parenthèses pour une seule expression.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des MERGE instructions INSERT ON CONFLICT dans PostgreSQL.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'une erreur d'analyse qui s'est produite pour les packages de procédures.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 5072 lors de la conversion des packages.
Oracle DW	Amazon Red	Correction d'une erreur en raison de laquelle le pack d'extension AWS SCT n'était pas appliqué lors de l'application du code converti à la base de données cible.
Oracle DW	Amazon Red	Correction d'une erreur qui AWS SCT empêchait d'appliquer certains fichiers du pack d'extension lors de l'utilisation de l'assistant du pack d'extensions.
Oracle DW	Amazon Red	Résolution d'un problème en AWS SCT raison duquel il était impossible de procéder à la migration AWS Snowball Edge des données vers plus de 500 tâches exécutées en parallèle.
Oracle DW	Amazon Red	Résolution d'un problème de conversion incorrecte des fonctions définies par l'utilisateur avec des types définis par l'utilisateur.

## Notes de mise à jour pour la AWS SCT version 669

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Amélioration du processus d'évaluation multiserveur, qui permet de déterminer la plate-forme de base de données cible optimale pour vos bases de données sources. Désormais, AWS SCT ignore



Source	Cible	Nouveautés, améliorations ou correctifs
		la AWS Secrets Manager clé si vous fournissez les informations d'identification de la base de données dans le fichier de valeurs séparées par des virgules (CSV) d'entrée. Pour de plus amples informations, veuillez consulter <a href="#">Rapport d'évaluation multiserveur</a> .
Tous	Tous	Résolution d'un problème selon lequel le rapport d'évaluation multiserveur incluait l'adresse IP de votre base de données source lorsque vous utilisiez un formulaire secret AWS Secrets Manager pour vous connecter à la base de données.
Tous	Amazon Red	Configuration automatique mise en œuvre des paramètres de la machine virtuelle Java (JVM) en fonction du système d'exploitation et de la RAM disponible. AWS SCT utilise cette JVM pour exécuter le travail des agents d'extraction de données.
Tous	Amazon Red	Résolution d'un problème en raison duquel les agents d'extraction de données ne démarraient pas dans Ubuntu.
Tous	Amazon Red	Résolution d'un problème en raison duquel les tâches d'extraction de données ne démarraient pas après l'exécution du <code>StartAgent.bat</code> fichier sous Windows.
Base de données Azure SQL	Aurora PostgreSQL	Résolution d'un problème de conversion incorrecte des noms de colonnes lorsque l'option Générer des noms uniques pour les index était activée.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Red	Implémentation de la conversion des fonctions VOID renvoyant aux procédures.

Source	Cible	Nouveautés, améliorations ou correctifs
Greenplum	Amazon Red	Résolution d'un problème d'échec de la migration de données lorsque la base de données source incluait des valeurs non numériques (NaN) dans des colonnes numériques. AWS SCT les agents d'extraction de données remplacent désormais les valeurs NaN par des valeurs NULL.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Ajout d'un nouveau paramètre de conversion pour spécifier les TIME FORMAT options DATE FORMAT et lors de la conversion des fonctions CHAR intégrées.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Ajout d'une action 8534 pour la conversion des curseurs prédéfinis déclarés avec la WITHOUT RETURN clause. Si votre curseur ne renvoie aucun ensemble de résultats, AWS SCT attribue une NULL valeur au nom de votre curseur dans le code converti et déclenche une action.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Modification de la CURRENT CLIENT_APPLNAME propriété identifiante AWS SCT lors de la connexion à la base de données source.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation d'un nouveau paramètre de conversion pour spécifier les TIME FORMAT options DATE FORMAT et lors de la conversion des fonctions CHAR intégrées.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion LEAVE des instructions en instructions par BEGIN . . . END blocs.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion de XMLPARSEXMLTABLE, et XMLNAMESPACES des fonctions.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des fonctions CHAR intégrées.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des curseurs.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion des instructions de FOR boucle.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Conversion améliorée de l'utilisation des types de tables dans SELECT les relevés.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Support implémenté de la nouvelle version 2.2.0 du fichier de configuration des fonctionnalités de Babelfish. Ce fichier définit les fonctionnalités SQL qui sont prises en charge et non prises en charge par des versions spécifiques de Babelfish.
Netezza	Amazon Redshift	Agents d'extraction de données améliorés pour résoudre un problème où aucune ligne n'était supprimée de la table cible pendant la réplication des données en cours.
Oracle	Amazon RDS for Oracle	Conversion améliorée des fonctionnalités d'Oracle Database Enterprise Edition.
Oracle	Aurora PostgreSQL PostgreSQL	Conversion des GROUPING_ID fonctions implémentée.
Oracle	Aurora PostgreSQL PostgreSQL	Conversion de code SQL améliorée dans les applications C# en ajoutant la prise en charge du mappage personnalisé des types de données en mode interface de ligne de commande (CLI).

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des tables imbriquées pour éviter une action inattendue 9996.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte de l'appel d'un constructeur d'objet.
Oracle DW	Amazon Red	Support implémenté des partitions de table existantes pour la migration des données. Pour accélérer la migration des données, AWS SCT crée des sous-tâches pour chaque partition de la table source qui n'est pas vide. Pour de plus amples informations, veuillez consulter <a href="#">Utilisation du partitionnement natif</a> .
Teradata	Amazon Red	Conversion améliorée des CAST fonctions avec <code>TIME WITH TIME ZONE AS TIMESTAMP</code> , <code>TIME WITH TIME ZONE AS CHAR</code> , et des <code>TIMESTAMP AS TIME WITH TIME ZONE</code> arguments.
Teradata	Amazon Red	Conversion améliorée des CAST fonctions grâce à l' <code>FORMAT</code> option.
Teradata	Amazon Red	Résolution d'un problème en raison duquel <code>CEIL</code> les fonctions n'étaient pas converties.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des <code>MERGE</code> instructions contenant des <code>DELETE</code> clauses.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des <code>TO_CHAR</code> fonctions avec des arguments de date et de format.

## Notes de mise à jour pour la AWS SCT version 668

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Amazon Red	Résolution d'un problème en raison duquel les opérateurs de multiplication dans les règles de migration ne fonctionnaient pas correctement. Ces opérateurs permettent de modifier la longueur de <code>dechar</code> , <code>varcharnvarchar</code> , et les types de <code>string</code> données. Pour de plus amples informations, veuillez consulter <a href="#">Création de règles de migration</a> .
Analyses Azure Synapse	Amazon Red	Support implémenté des <code>CONVERT</code> fonctions avec <code>VARCHAR</code> arguments.
Analyses Azure Synapse	Amazon Red	Amélioration de la conversion des <code>SELECT</code> déclarations contenant <code>NOLOCK</code> des clauses.
Analyses Azure Synapse	Amazon Red	Conversion améliorée des <code>UPDATE</code> instructions comportant des alias ou des <code>FROM</code> clauses <code>SET</code> et.
Greenplum	Amazon Red	Partitionnement virtuel automatique mis en œuvre pour la migration des données. AWS SCT utilise la colonne <code>GP_SEGMENT_ID</code> système pour créer des partitions.
Greenplum	Amazon Red	Support mis en œuvre <code>RETURN QUERY</code> et <code>RETURN SETOF</code> clauses.
Greenplum	Amazon Red	Support implémenté des <code>SUBSTRING</code> fonctions avec trois paramètres.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des <code>SUBSTR</code> fonctions avec des <code>LOCATE</code> paramètres.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Ajout d'une option permettant d'utiliser un tableau de REFCURSOR variables pour renvoyer des ensembles de résultats dynamiques. Lorsque vous sélectionnez cette option dans les paramètres de conversion, un OUT paramètre supplémentaire est AWS SCT ajouté au code converti.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Support implémenté des instructions de FOR boucle.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Support des XMLPARSE fonctions mis en œuvre. Ajout d'un élément d'action 8541 pour les bandes d'espaces blancs dans les XMLPARSE fonctions.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion de plusieurs gestionnaires d'exceptions dans un seul BEGIN . . . END bloc.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion INSERT et DELETE déclencheurs améliorés.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des appels de procédure imbriqués.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des types de tables.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème en raison duquel les NOT opérations logiques au niveau du bit étaient incorrectement converties pour des valeurs entières.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème en raison duquel les tableaux locaux n'étaient pas initialisés dans PostgreSQL version 8.0.2 ou antérieure.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des MERGE instructions contenant des WHEN NOT MATCHED BY SOURCE clauses.



Source	Cible	Nouveautés, améliorations ou correctifs
MySQL	Aurora MySQL	Résolution d'un problème lié à la détermination AWS SCT incorrecte des autorisations utilisateur accordées par le <code>rds_superuser</code> rôle.
Netezza	Amazon Red	Amélioration du chargeur de métadonnées source pour garantir le chargement AWS SCT correct des objets de base de données dont les noms sont en minuscules.
Oracle	Aurora PostgreSQL L  PostgreSQL L	De nouvelles fonctions ont été ajoutées au pack d'extension pour améliorer la conversion des tables imbriquées locales. Ces nouvelles fonctions émulent <code>PRIOR</code> , <code>NEXT</code> , <code>LIMIT</code> , <code>FIRST</code> , <code>LAST</code> , <code>EXISTS</code> , <code>EXTEND TRIMDELETE</code> , et <code>SET</code> les fonctions de votre code Oracle source. Pour de plus amples informations, veuillez consulter <a href="#">Packs d'extension</a> .
Oracle	Aurora PostgreSQL L  PostgreSQL L	Ajout de la possibilité de spécifier l'étendue de conversion pour les applications C#. Les utilisateurs peuvent désormais exclure les sous-ensembles du projet d'application source de la zone de conversion.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Mise en œuvre du support <code>COUNT</code> des méthodes dans les collections.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Support implémenté des variables et des constructeurs dans les tables imbriquées.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Support <code>RATIO_TO_REPORT</code> et <code>STANDARD_HASH</code> fonctions mis en œuvre.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des objets volumineux (LOBs) dans le cadre du pack d' AWS SCT extension.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des collections locales.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des <code>JOIN</code> instructions contenant des <code>USING</code> clauses dans lesquelles les noms de colonnes n'incluent pas le nom de la table.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion <code>EMPTY_BLOB</code> et <code>EMPTY_CLOB</code> fonctions mises en œuvre.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des variables de liaison positionnelles dans les applications C#.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Mise en œuvre de la conversion des déclencheurs multi-événements.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des déclencheurs récursifs.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des déclencheurs avec la variable @@rowcount globale.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des fonctions d'agrégation dans la SET clause des UPDATE instructions.

Source	Cible	Nouveautés, améliorations ou correctifs
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème d'apparition inattendue de l'action 42702 lors de la conversion des UPDATE instructions.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des CONVERT fonctions comportant des CHAR arguments.
Snowflake	Amazon Red	Ajout de la prise en charge de Snowflake en tant que source de migration de données avec des agents d'extraction de AWS SCT données. Pour de plus amples informations, veuillez consulter <a href="#">Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool</a> .
Teradata	Amazon Red	Conversion améliorée des CAST fonctions avec des TIMESTAMP AS TIME WITH TIMEZONE arguments.

## Notes de mise à jour pour la AWS SCT version 667

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Support implémenté des scripts d'extraction, de transformation et de chargement (ETL) Informatica en mode interface de ligne de commande (CLI). AWS SCT redirige automatiquement vos scripts ETL Informatica vers la nouvelle base de données cible. AWS SCT Convertit également les noms d'objets et le code SQL intégré à vos objets Informatica. Pour de plus amples informations, veuillez consulter <a href="#">Scripts ETL Informatica</a> .

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Amazon Redshift	La version minimale du pilote prise en charge pour Amazon Redshift a été augmentée à 2.1.0.9. Pour de plus amples informations, veuillez consulter <a href="#">Installation des pilotes JDBC pour AWS Schema Conversion Tool</a> .
Analyses Azure Synapse	Amazon Redshift	Ajout d'une nouvelle fonction au pack d'extension pour améliorer la conversion de la CONVERT fonction avec trois arguments de date et d'heure.
Analyses Azure Synapse	Amazon Redshift	Conversion améliorée de la DATEDIFF fonction.
Analyses Azure Synapse  Microsoft SQL Server DW	Amazon Redshift	Mise à jour de la version du pack d'extension. Assurez-vous d'appliquer la dernière version du pack d'extension à vos AWS SCT projets existants. Pour de plus amples informations, veuillez consulter <a href="#">Packs d'extension</a> .
BigQuery	Amazon Redshift	Résolution d'un problème en raison duquel les objets filtrés n'étaient pas convertis en mode interface de ligne de commande (CLI).
Greenplum	Amazon Redshift	Correction d'une erreur AWS SCT qui empêchait de convertir les tables temporaires déclarées dans une procédure stockée.
Greenplum	Amazon Redshift	Correction d'une erreur en raison de l'absence d'attributs de codage de colonne dans le code converti.
Microsoft SQL Server	Aurora PostgreSQL  PostgreSQL	Implémentation de la conversion UPDATE des instructions pour les tables autoréférencées contenant plusieurs INNER JOIN clauses.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Support implémenté <code>inserted</code> et tables <code>deleted</code> temporaires que SQL Server utilise pour les déclencheurs DML.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Conversion améliorée des types définis par l'utilisateur dans les procédures stockées créées dans différents schémas de base de données. Problème résolu : AWS SCT impossible de trouver le type de données et affichage d'une action 9996.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Résolution d'un problème en raison duquel des crochets ( <code>[ ]</code> ) apparaissaient de manière inattendue autour des noms d'objets de base de données dans le code converti.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Résolution d'un problème de conversion incorrecte des <code>@ROWCOUNT</code> fonctions.
Microsoft SQL Server DW	Amazon Redshift	Support mis en œuvre pour les types de <code>geography</code> données <code>geometry</code> et les types de données.
Microsoft SQL Server DW	Amazon Redshift	Support implémenté du <code>MAX</code> mot clé dans les déclarations de type de données dans le code converti.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server DW	Amazon Redshift	Conversion améliorée des DATEADD fonctions.
Oracle	Aurora PostgreSQL PostgreSQL	Amélioration de la conversion du code SQL dans les applications Java en ajoutant la prise en charge du MyBatis framework. Pour de plus amples informations, veuillez consulter <a href="#">Code SQL en Java</a> .
Oracle	Aurora PostgreSQL PostgreSQL	Conversion de code SQL améliorée dans les applications Java qui utilisent le MyBatis framework. Ajout d'une action 30411 pour le code SQL dont la syntaxe n'est pas prise en charge.
Oracle	Aurora PostgreSQL PostgreSQL	Conversion de code SQL améliorée dans les applications Pro*C en ajoutant la prise en charge des typedef struct déclarations.
Oracle	Aurora PostgreSQL PostgreSQL	Support CROSS JOIN et LEFT JOIN déclarations mis en œuvre.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des MERGE relevés. Résolution d'un problème en raison duquel les valeurs à insérer manquaient dans le code converti.
Teradata	Amazon Red	Modification des paramètres de codage de compression de colonne par défaut AWS SCT utilisés dans le code converti pour qu'ils correspondent aux paramètres par défaut d'Amazon Redshift. Pour plus d'informations, consultez la section <a href="#">Encodages de compression</a> dans le manuel Amazon Redshift Database Developer Guide.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des opérations mathématiques utilisant le type de TIME données.
Teradata	Amazon Redshift RSQL	Implémentation de la conversion FastExport du code contenu dans les scripts shell.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'une erreur en raison de laquelle les %data instructions AWS SCT n'étaient pas COALESCE converties.
Vertica	Amazon Red	Suggestions d'optimisation des conversions améliorées lorsqu'un utilisateur sélectionne une stratégie d'optimisation.

## Notes de mise à jour pour la AWS SCT version 666

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Azure SQL	Aurora PostgreSQL L	Résolution d'une erreur d'analyse qui se produisait pour les ON clauses contenues dans des JOIN instructions.



Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	PostgreSQL	
Analyses Azure Synapse	Amazon Red	Trois nouvelles fonctions ont été ajoutées au pack d'extension pour améliorer la conversion de la CONVERT fonction avec des arguments de date et d'heure.
Analyses Azure Synapse	Amazon Red	Amélioration du chargeur de métadonnées source pour garantir le chargement AWS SCT des schémas de base de données du système.
Analyses Azure Synapse	Amazon Red	Correction d'une erreur de résolution survenue pour les colonnes des tables temporaires.
Analyses Azure Synapse	Amazon Red	Implémentation de la conversion BINARY et VARBINARY des types de données vers le type de VARBYTE données.
Analyses Azure Synapse	Amazon Red	Support implémenté du type de TIME données dans le code converti.
Analyses Azure Synapse	Amazon Red	Amélioration de la conversion des COLLATE clauses. Résolution d'un problème en raison duquel l'action 31141 apparaissait de manière inattendue lors de la conversion de colonnes avec le classement de base de données par défaut.
BigQuery	Amazon Red	Conversion implémentée des procédures qui modifient les paramètres d'entrée.
Greenplum	Amazon Red	Résolution d'un problème lié à l' AWS SCT utilisation d'une requête incompatible avec les bases de données Greenplum 6.x.

Source	Cible	Nouveautés, améliorations ou correctifs
IBM DB2 pour z/OS	Aurora PostgreSQL PostgreSQL	Amélioration de la conversion des EXCEPTION sections en transférant les gestionnaires d'exceptions de DB2 pour z/OS vers PostgreSQL.
IBM DB2 pour z/OS	Aurora PostgreSQL PostgreSQL	Conversion améliorée des OPEN CURSOR relevés.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Implémentation de la conversion des IIF fonctions à l'aide d'CASEexpressions.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Résolution d'un problème en raison duquel les procédures comportant des paramètres à valeur tabulaire étaient mal converties lorsque l'CREATE PROCEDURE instruction n'incluait aucun bloc. BEGIN . . . END
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Résolution d'un problème de conversion incorrecte de la SCOPE_IDENTITY fonction.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Amazon RDS for Oracle	Correction d'une erreur de chargement qui se produisait avec le <code>SELECT_CATALOG_ROLE</code> rôle lors de l'utilisation d'Oracle 10g comme source.
Oracle	Amazon RDS for Oracle	Le chargeur a été amélioré pour prendre en charge les tâches Oracle Scheduler.
Oracle	Aurora PostgreSQL PostgreSQL	Implémentation de la conversion JOIN des instructions avec USING des clauses.
Oracle	Aurora PostgreSQL PostgreSQL	Amélioration des performances du code converti lorsque le code source inclut des variables globales dans les WHERE clauses.
Oracle	Aurora PostgreSQL PostgreSQL	Amélioration de la conversion du code SQL dans les applications Java en ajoutant la prise en charge du MyBatis framework. Pour de plus amples informations, veuillez consulter <a href="#">Code SQL en Java</a> .
Oracle DW	Amazon Redshift	Mise en œuvre de la conversion des opérateurs UNPIVOT relationnels PIVOT et.
Teradata	Amazon Redshift	Correction d'une erreur à cause de laquelle le code source utilisant des objets JSON n'était pas converti.
Teradata	Amazon Redshift	Correction d'une erreur à cause de laquelle les tables créées par un utilisateur abandonné n'étaient pas correctement chargées.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Implémentation de la conversion des INSTR fonctions en fonctions natives d'Amazon RedshiftSTRPOS.
Teradata	Amazon Red	Conversion implémentée des TRANSLATE fonctions NVP et.
Teradata	Amazon Red	Conversion améliorée des COALESCE expressions.
Teradata	Amazon Red	Conversion améliorée des DECLARE CONDITION relevés.
Teradata	Amazon Red	Conversion améliorée des EXTRACT fonctions avec l'élément de SECOND syntaxe.
Teradata	Amazon Red	Conversion améliorée des SQLCODE variables SQLSTATE et dans les LOOP instructions.
Teradata	Amazon Red	Conversion améliorée des index uniques.
Teradata	Amazon Red	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion d'CURRENT_TIMESTAMP instructions avec une précision fractionnaire définie sur 3.
Teradata	Amazon Red	Résolution d'un problème en raison duquel les barres obliques inversées étaient incorrectement converties en chaînes littérales.
Teradata	Amazon Red	Résolution d'un problème en raison duquel EXEC les instructions converties incluaient un nom de champ incorrect dans l'ADD CONSTRAINT instruction.
Teradata	Amazon Red	Résolution d'un problème en raison duquel les QUALIFY sous-requêtes converties contenaient un nom de sous-requête incorrect.
Teradata	Amazon Red	Résolution d'un problème en raison duquel les vues converties n'étaient pas appliquées. Ajout d'une conversion explicite vers un type de données spécifique pour les valeurs NULL dans le code converti.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des fonctions de date et d'heure.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Résolution d'un problème en raison duquel les chaînes littérales hexadécimales n'étaient pas converties.

## Notes de mise à jour pour la AWS SCT version 665

Source	Cible	Nouveautés, améliorations ou correctifs
Analyses Azure Synapse	Amazon Red	Implémentation de la conversion des CONCAT fonctions avec VARCHAR des arguments.
Analyses Azure Synapse	Amazon Red	Conversion améliorée des CREATE TABLE instructions qui créent des tables temporaires et n'incluent pas le nom du schéma. AWS SCT crée le dbo schéma pour stocker ces tables temporaires dans la base de données cible.
Analyses Azure Synapse	Amazon Red	Conversion améliorée des DROP TABLE instructions que vous exécutez sur des tables temporaires.
Analyses Azure Synapse	Amazon Red	Conversion améliorée des OBJECT_ID instructions avec les BEGIN...END blocs.
Analyses Azure Synapse	Amazon Red	Résolution d'une erreur qui AWS SCT empêchait la conversion des procédures stockées contenant des commentaires en bloc.
BigQuery	Amazon Red	Mise en œuvre de la conversion BigQuery des entrepôts de données vers Amazon Redshift. Pour de plus amples informations, veuillez consulter <a href="#">Connexion à Google BigQuery avec AWS Schema Conversion Tool</a> .

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Conversion améliorée des déclencheurs qui gèrent plusieurs événements et fonctionnent avec <code>inserted</code> les tables <code>deleted</code> système de SQL Server.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Correction d'une erreur de résolution qui s'est produite pour les tables <code>deleted</code> système <code>inserted</code> et dans SQL Server.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Support implémenté de la nouvelle version 2.1.0 du fichier de configuration des fonctionnalités de Babelfish. Ce fichier définit les fonctionnalités SQL qui sont prises en charge et non prises en charge par des versions spécifiques de Babelfish.
Oracle	Aurora MySQL MariaDB MySQL	Résolution d'un problème de conversion incorrecte du type de <code>varchar2</code> données.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora MySQL	Pour les bases de données Oracle version 12c et supérieures, AWS SCT prend en charge les types de données étendus suivants :
	Aurora PostgreSQL L	<ul style="list-style-type: none"> <li>• VARCHAR2</li> <li>• NVARCHAR2</li> <li>• RAW</li> </ul>
	MariaDB MySQL PostgreSQL L	AWS SCT a augmenté la longueur de colonne maximale prise en charge de 8 000 à 32 767 octets pour ces types de données.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'une erreur d'analyse qui s'est produite pour le package Oracle Event Processing.
Teradata	Amazon Red	Ajout d'une action 13214 pour plusieurs RESET WHEN clauses dans une seule SELECT instruction.
Teradata	Amazon Red	Ajout d'une action pour les SQLSTATE variables situées en dehors d'un bloc de gestion des exceptions.
Teradata	Amazon Red	Implémentation de la conversion ACTIVITY_COUNT des variables en ROW_COUNT .
Teradata	Amazon Red	Conversion implémentée de la ST_TRANSFORM fonction de géométrie intégrée.
Teradata	Amazon Red	Amélioration de la conversion des instructions de suppression en vues sans la WHERE clause.
Teradata	Amazon Red	Conversion améliorée des CAST opérateurs dans les expressions.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Amélioration de la conversion des GROUP BY clauses.
Teradata	Amazon Red	Conversion améliorée INSTR et fonctions REGEXP_INSTR intégrées.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des références d'alias de colonnes latérales.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte des noms de colonnes dans la QUALIFY sous-requête.
Teradata	Amazon Red	Implémentation de la conversion des .QUIT commandes avec le mot clé ERRORCODE status value.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'un problème d'apparition inattendue de l'action 9996 lors de la conversion des CREATE instructions.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'un problème d'apparition inattendue de l'action 9998 lors de la conversion des END instructions.

## Notes de mise à jour pour la AWS SCT version 664

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la prise en charge d'Amazon Redshift Serverless en tant que source et cible pour les projets de migration de bases de données dans. AWS SCT Pour vous connecter à Amazon Redshift Serverless, assurez-vous d'utiliser le pilote Amazon Redshift JDBC version 2.1.0.9 ou supérieure.
Tous	Tous	Amélioration de l'interface utilisateur de la fenêtre des paramètres de conversion. AWS SCT affiche désormais les paramètres uniquement pour les paires de conversion de base de données avec des règles



Source	Cible	Nouveautés, améliorations ou correctifs
		de mappage créées. Pour de plus amples informations, veuillez consulter <a href="#">Cartographie des types de données</a> .
Tous	Tous	Le rapport d'évaluation a été mis à jour afin de supprimer les informations dupliquées concernant la ligne et la position de l'élément d'action.
Tous	Amazon Red	Implémentation de l'équilibrage automatique de la mémoire dans les tâches d'extraction de données.
Tous	Amazon Red	Résolution d'une erreur empêchant les agents d'extraction de données de se connecter aux AWS Snowball Edge appareils.
Base de données Azure SQL	Aurora MySQL	Support implémenté de SUSE Linux 15.3 en tant que plate-forme pour exécuter des agents d'extraction de données.
IBM DB2 pour z/OS	Aurora PostgreSQL	
IBM Db2 LUW	MySQL	
Microsoft SQL Server	PostgreSQL	
MySQL	L	
Oracle		
PostgreSQL		
SAP ASE		

Source	Cible	Nouveautés, améliorations ou correctifs
Analyses Azure Synapse	Amazon Red	Conversion améliorée des DATEADD fonctions.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Ajout de la possibilité de modifier le classement des colonnes dans les règles de migration.
Microsoft SSIS	AWS Glue  AWS Glue Studio	Résolution d'une erreur inattendue qui se produisait lorsque les utilisateurs sélectionnaient un script source.
Oracle	Aurora MySQL  MariaDB  MySQL	Implémentation de la conversion de l'utilisation des fonctions stockées sous forme d'expressions de colonne générées. AWS SCT crée des déclencheurs pour émuler ce comportement car MySQL ne prend pas en charge l'utilisation de fonctions stockées en tant qu'expressions de colonne générées.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Implémentation de la conversion des fonctions UTL_MATCH du package dans le cadre du pack d' AWS SCT extension.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion implémentée de la REGEXP_LIKE fonction avec le NULL paramètre.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée de la SYS_EXTRACT_UTC fonction.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion du code SQL dans les applications C++ grâce à la mise en œuvre de la prise en charge des Wcsncats fonctions WcscatsWcscpys, et. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de code SQL dans des applications C++ avec AWS Schema Conversion Tool</a> .
Oracle DW  Snowflake	Amazon Redshift	Résolution d'un problème en raison duquel les instructions converties n'incluaient pas de conversion explicite des valeurs vers le type de données de colonne. Ce problème s'est produit dans les instructions qui utilisent les résultats de requêtes provenant d'autres tables.
Teradata	Amazon Redshift	Ajout de la possibilité de modifier le classement des colonnes entre case sensitive et case insensitive dans les règles de migration. Pour de plus amples informations, veuillez consulter <a href="#">Appliquer les règles de migration</a> .
Teradata	Amazon Redshift	Correction d'une erreur de résolution qui se produisait pour les CREATE TABLE AS instructions.
Teradata	Amazon Redshift	Correction d'une erreur en raison de laquelle la P_INTERSECT fonction intégrée avec une COALESCE expression n'était pas convertie.
Teradata	Amazon Redshift	Implémentation de la conversion des colonnes nommées OID_OID afin d'éviter l'utilisation d'un mot clé réservé dans Amazon Redshift.
Teradata	Amazon Redshift	Implémentation de la conversion RENAME des instructions pour les fonctions, les procédures, les vues et les macros.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Implémentation de la STROKE conversion de la SPLIT_PART fonction en fonction dans Amazon Redshift.
Teradata	Amazon Red	Conversion améliorée des fonctions INSTR et REGEXP_INSTR du système.
Teradata	Amazon Red	Conversion améliorée du type TIME de données.
Teradata	Amazon Red	Émulation améliorée des MULTISSET tables SET et en implémentant la conversion des index uniques principaux et secondaires.
Teradata	Amazon Red	Résolution d'une erreur d'analyse qui s'est produite pour la CHARACTER fonction.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'une erreur survenue lorsque les utilisateurs supprimaient les scripts Teradata Basic Teradata Query (BTEQ) du projet. AWS SCT

## Notes de mise à jour pour la AWS SCT version 663

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la possibilité de modifier la longueur de char, varcharnvarchar, et les types de string données à l'aide de l'opérateur de multiplication dans une règle de migration. Pour de plus amples informations, veuillez consulter <a href="#">Appliquer les règles de migration</a> .
Tous	Tous	Mise en œuvre de la prise en charge de trois nouvelles colonnes dans le rapport d'évaluation multiserveur et mise à jour du format du fichier d'entrée. Assurez-vous d'utiliser le modèle mis à jour du fichier d'entrée avec la dernière version de AWS SCT. Pour de plus amples informations, veuillez consulter <a href="#">Création d'un rapport d'évaluation multiserveur dans AWS Schema Conversion Tool</a> .

Source	Cible	Nouveautés, améliorations ou correctifs
Analyses Azure Synapse	Amazon Red	Conversion améliorée des OBJECT_ID relevés.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Ajout de la prise en charge de Babelfish for Aurora PostgreSQL 1.2.0 en tant que plate-forme cible pour les rapports d'évaluation de la migration des bases de données. Pour plus d'informations, consultez la section <a href="#">Fonctionnalités prises en charge dans Babelfish par version</a> dans le guide de l'utilisateur Amazon Aurora.
Microsoft SQL Server DW	Amazon Red	Ajout du support pour les AT TIME ZONE clauses.
Microsoft SQL Server DW	Amazon Red	Résolution d'un problème de conversion incorrecte d'une instruction en dehors du BEGIN/END bloc.
Netezza	Amazon Red	Conversion améliorée du type de TIME données et mise en œuvre de la conversion des fonctions, expressions et littéraux intégrés associés.
Oracle	Aurora PostgreSQL  PostgreSQL	Correction d'une erreur de chargement survenue lors de l'utilisation d'Oracle 10g comme source.
Oracle	Aurora PostgreSQL  PostgreSQL	Conversion améliorée des FETCH clauses OFFSET et.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des procédures comportant des OUT paramètres avec des valeurs par défaut.
Oracle DW	Amazon Red	Conversion améliorée des fonctions Oracle en fonctions définies par l'utilisateur Amazon Redshift.
Snowflake	Amazon Red	Amélioration de la conversion des WITH clauses.
Teradata	Amazon Red	Ajout d'une nouvelle action 13209 pour les caractères multi-octets non pris en charge pour le CHAR type de données.
Teradata	Amazon Red	Correction d'une erreur de chargement à cause de laquelle les tables n'étaient pas complètement chargées.
Teradata	Amazon Red	Correction d'une erreur de transformateur qui empêchait la conversion de la P_INTERSECT fonction intégrée d'une JOIN condition.
Teradata	Amazon Red	Correction d'un problème en raison duquel le nom d'une vue était converti dans le mauvais cas lorsque l'SELECT instruction était exécutée sur une table contenant des caractères spéciaux dans son nom.
Teradata	Amazon Red	Conversion améliorée des INSERT instructions avec la UNTIL_CHANGED valeur dans le type de PERIOD(DATE) données.
Teradata	Amazon Red	Conversion améliorée de la FORMAT fonction intégrée à l'aide de la TO_CHAR fonction dans Amazon Redshift.
Teradata	Amazon Red	Conversion améliorée de la RANK fonction intégrée pour garantir que le code converti renvoie des valeurs NULL dans le même ordre que le code source.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Conversion améliorée des contraintes uniques telles que les index uniques principaux ou secondaires.

## Notes de mise à jour pour la AWS SCT version 662

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la possibilité de créer automatiquement des AWS SCT projets pour chaque base de données source lors de la création du rapport d'évaluation multiserveur. Lorsque cette option est activée, AWS SCT vous pouvez ajouter des règles de mappage à ces projets et enregistrer des statistiques de conversion pour une utilisation hors ligne. Pour de plus amples informations, veuillez consulter <a href="#">Création d'un rapport d'évaluation multiserveur dans AWS Schema Conversion Tool</a> .
Tous	Tous	Prise en charge du pourcentage (%) en tant que caractère générique dans les noms de base de données et de schéma lors de la création du rapport d'évaluation multiserveur.
Tous	Aurora MySQL  Aurora PostgreSQL	Mise à jour du runtime de toutes les AWS Lambda fonctions vers la version 3.9 de Python.
Tous	Amazon Red	Tous les agents d'extraction de données à utiliser ont été mis à niveau AWS SDK for Java 2.x.
Base de données Azure SQL	Aurora PostgreSQL	Amélioration de la conversion des DELETE déclarations contenant NON EXISTS des clauses.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	PostgreSQL	
Analyses Azure Synapse	Amazon Red	Résolution d'une erreur entraînant l'échec de la connexion à une base de données source.
IBM DB2 pour z/OS	Aurora PostgreSQL L  PostgreSQL L	Résolution d'une erreur à cause de laquelle le code converti d'un déclencheur incluait deux mentions de l'alias de l'objet.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Amélioration de la conversion des objets dont les noms sont écrits en majuscules et minuscules lorsque l'option Traiter le nom de l'objet de base de données en distinguant majuscules et minuscules est activée.
Microsoft SQL Server DW  Teradata	Amazon Red	Mise en œuvre de la conversion des opérateurs UNPIVOT relationnels PIVOT et.
Netezza	Amazon Red	Conversion du type de TIME données implémentée.



Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora MySQL  Aurora PostgreSQL L  MySQL  PostgreSQL L	Implémentation de la conversion constante UTL_TCP.CRLF du package.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Correction d'un problème lié au pack d'extension en raison duquel la longueur des types de données pour les colonnes de longueur variable n'était pas maintenue pendant la conversion.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion de code SQL implémentée dans les applications C++. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de code SQL dans des applications C++ avec AWS Schema Conversion Tool</a> .
Oracle	Aurora PostgreSQL L  PostgreSQL L	Support implémenté de la distinction majuscules/minuscules pour la conversion de variables globales et de tableaux associatifs.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des TO_NUMBER fonctions TO_CHAR, TO_DATE, et dans le pack d'extension.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée de l'TABLE ( ) opérateur.
Oracle DW	Amazon Red	Ajout du support pour la conversion des clés primaires et d'autres contraintes.
Oracle DW	Amazon Red	Correction d'un problème en raison duquel l'action 12054 n'apparaissait pas lors de la conversion des instructions conditionnelles.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Résolution d'une erreur lors de la création d'un objet avec un nom vide dans l'arborescence cible lors de la conversion de tables contenant des colonnes de type défini par l'utilisateur.
SAP ASE	Aurora PostgreSQL L  PostgreSQL L	Correction d'une erreur de chargement pour les objets stockés tels que les scripts, les routines, etc.
Snowflake	Amazon Red	Correction d'un problème en raison duquel l'action 22152 n'apparaissait pas lorsque cela était nécessaire et AWS SCT affichait le résultat de la conversion sous forme de commentaire.

Source	Cible	Nouveautés, améliorations ou correctifs
Snowflake	Amazon Red	Conversion améliorée des fonctions de date et d'heure ; prise en charge des fuseaux horaires mise en œuvre.
Snowflake	Amazon Red	Résolution d'un problème en raison duquel les expressions de table communes non récursives (CTEs) contenant une WITH clause étaient converties en expressions récursives. CTEs
Teradata	Amazon Red	Conversion améliorée des UPDATE instructions avec des liens de table en condition.
Teradata	Amazon Red	Conversion améliorée des RENAME TABLE relevés.
Teradata	Amazon Red	Résolution d'un problème en raison duquel des colonnes vides apparaissaient dans le fichier de valeurs séparées par des virgules (CSV) contenant un rapport d'évaluation.
Teradata	Amazon Redshift RSQL	Correction d'une erreur en raison de l'absence d'un point-virgule à la fin de la macro BTEQ (Basic Teradata Query) convertie.
Teradata	Amazon Redshift RSQL	Conversion améliorée de plusieurs valeurs de types de données dans CASE les instructions.
Teradata	Amazon Redshift RSQL	Amélioration de la conversion de la LIKE ANY clause avec un ESCAPE caractère.
Teradata	Amazon Redshift RSQL	Conversion améliorée de la CAST fonction dans les INSERT instructions.
Teradata	Amazon Redshift RSQL	Conversion améliorée des fuseaux horaires, mise en œuvre de la cartographie des zones de fuseaux horaires.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift RSQL	Résolution d'un problème d'apparition inattendue de l'action 9998 lors de la conversion de scripts shell avec des scripts BTEQ.
Teradata	Amazon Redshift RSQL AWS Glue	Implémentation de la limite de 500 caractères pour les valeurs des variables de substitution.
Vertica	Amazon Red	Implémentation de la conversion des types de RAW données BINARY VARBINARY LONG BINARYBYTEA,,, et en type de VARBYTE données.
Vertica	Amazon Red	Conversion améliorée des fonctions et des littéraux intégrés.

## Notes de mise à jour pour la AWS SCT version 661

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Filtres ajoutés pour rechercher des règles de mappage dans la vue cartographique. Lorsque vous appliquez un filtre, AWS SCT affiche les règles correspondant aux conditions de filtrage dans la liste des mappages de serveurs. Pour de plus amples informations, veuillez consulter <a href="#">Modification des mappages de types de données dans AWS Schema Conversion Tool</a> .
Tous	Tous	Mise à niveau d'Apache Log4j vers la version 2.17.1.
Tous	Amazon Red	Ajout de la prise en charge de la migration des données vers Amazon Redshift à l'aide de la ENCRYPTED clause contenue dans la COPY commande.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Amazon Red	Amélioration de l'API REST des agents d'extraction de données. L'API REST mise à jour ajoute la prise en charge de nouvelles propriétés telles que la clé de chiffrement, le type de chiffrement, etc.
Tous	Amazon Red	Rôle implémenté dans les agents d'extraction de données. Cette mise à jour améliore la distribution des sous-tâches et permet AWS SCT d'attribuer des tâches aux agents libres du rôle spécifié.
Tous	Amazon Red	Une vérification a été mise en œuvre pour vérifier que tous les composants requis sont installés avant que le pack d'extension ne soit appliqué à Amazon Redshift.
Analyses Azure Synapse  Microsoft SQL Server DW	Amazon Red	Conversion améliorée des fonctions <code>ERROR_LINE</code> , <code>ERROR_MESSAGE</code> , <code>ERROR_NUMBER</code> , <code>ERROR_PROCEDURE</code> , <code>ERROR_SEVERITY</code> , et <code>ERROR_STATE</code> du système pour la gestion des erreurs.
IBM DB2 pour z/OS	Aurora MySQL  Aurora PostgreSQL  MySQL  PostgreSQL	Ajout de la prise en charge d'IBM Db2 for z/OS version 12 en tant que source pour les projets de migration de base de données dans. AWS SCT Pour de plus amples informations, veuillez consulter <a href="#">Connexion à IBM DB2 pour z/OS</a> .
IBM Db2 LUW	Tous	Le chargeur de métadonnées source a été amélioré pour garantir le chargement AWS SCT des paramètres de routine qui dupliquent les noms de colonne.

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Microsoft Azure SQL Server	Aurora PostgreSQL	Correction d'une erreur de transformation pour les procédures utilisant l'instruction SET NOCOUNT ON set.
Microsoft SQL Server	PostgreSQL	
Base de données Microsoft Azure SQL Server	Aurora PostgreSQL	Conversion améliorée de la CONCAT fonction lorsqu'une valeur d'entrée est une variable du type défini par l'utilisateur.
Microsoft SQL Server	PostgreSQL	
Base de données Microsoft Azure SQL Server	Aurora PostgreSQL	Résolution d'un problème de conversion incorrecte de la DATEPART fonction.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	L	
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Support implémenté de la nouvelle version du fichier de configuration des fonctionnalités de Babelfish. Ce fichier définit les fonctionnalités SQL qui sont prises en charge et non prises en charge par des versions spécifiques de Babelfish.
Microsoft SQL Server DW	Amazon Redshift	Résolution d'un problème de conversion incorrecte des procédures comportant une EXECUTE instruction.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SSIS	AWS Glue	Amélioration de l'interface utilisateur de l'assistant de configuration des tâches. AWS SCT affiche désormais uniquement les connexions disponibles dans la section de configuration des connexions.
Microsoft SSIS	AWS Glue	Résolution d'un problème en raison duquel les règles de transformation n'étaient pas appliquées aux tâches de package et aux règles variables.
Microsoft SSIS	AWS Glue AWS Glue Studio	Ajout d'une nouvelle action 25042 pour les composants non pris en charge.
Microsoft SSIS	AWS Glue Studio	Mise en œuvre de la conversion des packages d'extraction, de transformation et de chargement (ETL) de Microsoft SQL Server Integration Services (SSIS) en AWS Glue Studio. Pour de plus amples informations, veuillez consulter <a href="#">SSIS vers AWS Glue Studio</a> .
Oracle	MariaDB	Correction d'un problème de conversion de l'MINUSopérateur.
Oracle	MariaDB	Conversion améliorée des ADD_MONTHS fonctionsROWNUM, SYS_GUIDTO_CHAR, et lorsque la variable sql_mode système dans MariaDB est Oracle.
Oracle	PostgreSQL	Ajout d'une option pour éviter la conversion des types de variables de liaison en types SQL dans les projets de conversion d'applications génériques.
Oracle	PostgreSQL	Ajout d'une option pour éviter d'ajouter le nom du schéma au nom de l'objet converti dans les projets de conversion d'applications génériques.
Oracle	PostgreSQL	Ajout du support du format de variable de ?x liaison pour la conversion du code SQL des applications.
Oracle DW	Amazon Red	Implémentation de la conversion du type de RAW données en type de VARBYTE données.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Ajout d'une option permettant d'émuler SET des tables dans le code converti. Pour cette émulation, AWS SCT supports MIN et MAX conditions.
Teradata	Amazon Red	Conversion améliorée des opérations de jointure comportant des paramètres de différents types de données. Cette mise à jour permet AWS SCT d'appliquer des règles de transformation lors de la conversion de telles opérations.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte de la GROUP BY clause.
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte de la QUALIFY clause.
Teradata	Amazon Red	Résolution d'une erreur inattendue survenue lors de l'importation FastExport des scripts.
Teradata	Amazon Redshift RSQL	Implémentation de la possibilité de modifier les valeurs des variables dans les scripts Teradata BTEQ et shell.
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel le script de manifeste était absent pour les FastLoad sessions Teradata converties.
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel l'extension du fichier manifeste était absente dans l'URL (Uniform Resource Locator) pour les FastLoad scripts convertis.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'une erreur de chargement pour les scripts contenant des variables de substitution.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'un problème en raison duquel l'action 27022 n'apparaissait pas lorsque cela était nécessaire.



## Notes de mise à jour pour la AWS SCT version 660

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la prise en charge du AWS Secrets Manager protocole SSL (Secure Sockets Layer) dans le rapport d'évaluation multiserveur. Pour de plus amples informations, veuillez consulter <a href="#">Création d'un rapport d'évaluation multiserveur dans AWS Schema Conversion Tool</a> .
Tous	Tous	Collecte de statistiques améliorée pour les objets convertis.
Tous	PostgreSQL L	Support implémenté de la version majeure de PostgreSQL 14 et de MariaDB 10.6 en tant que cibles de migration.
Analyses Azure Synapse	Amazon Redshift	Logique de transformation améliorée pour les noms des objets convertis.
Base de données Microsoft Azure SQL  Microsoft SQL Server	Aurora PostgreSQL L	Conversion améliorée du type XML de données.
Base de données Microsoft Azure SQL  Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème de conversion incorrecte des NOT LIKE clauses.

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Microsoft Azure SQL Server	Aurora PostgreSQL	Correction d'une erreur de transformation pour les procédures INSERT comportant UPDATE des instructions DELETE,, et qui incluent la OUTPUT clause.
Microsoft SQL Server	PostgreSQL	
Base de données Microsoft Azure SQL Server	Aurora PostgreSQL	Correction d'une erreur de transformation pour les procédures comportant l'RETURN @@ROWCOUNT instruction.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Tous	Conversion améliorée des procédures utilisant des serveurs liés.
Microsoft SQL Server	Tous	Ajout de la prise en charge de l'authentification Microsoft Windows dans le rapport d'évaluation multiserveur.
Microsoft SQL Server	Aurora PostgreSQL	Correction d'une erreur de transformation pour les constructeurs de valeurs de table.
	PostgreSQL	
Microsoft SQL Server DW	Amazon Redshift et AWS Glue	Conversion améliorée des scripts d'extraction, de transformation et de chargement (ETL) afin d'inclure le chemin correct vers les scripts convertis.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server DW	Amazon Red	Résolution d'un problème en raison duquel différents scripts convertis étaient générés pour les plateformes de base de données cibles virtuelles et réelles.
Oracle	PostgreSQL Aurora PostgreSQL	Ajout du support pour la conversion des index pour les vues matérialisées.
Oracle	PostgreSQL Aurora PostgreSQL	Correction d'un problème en raison duquel l'élément d'action 5982 n'apparaissait pas lors de la conversion PRIMARY KEY et UNIQUE des contraintes liées à l'NOVALIDATE option.
Oracle DW	Amazon Red	Résolution d'un problème d'affichage de catégories supplémentaires dans le schéma converti.
Teradata	Amazon Red	Correction d'un problème en raison duquel l'action 13185 n'apparaissait pas lors de la conversion d'une colonne non résolue en tant qu'argument de la CAST fonction.
Teradata	Amazon Red	Amélioration de la conversion DELETE et DELETE ALL des instructions permettant d'utiliser la TRUNCATE commande dans le code converti.
Teradata	Amazon Red	Conversion améliorée des SET tables.
Teradata	Amazon Red	Amélioration de la conversion de la NORMALIZE condition.
Teradata	Amazon Red	Mise à jour du rapport d'évaluation pour supprimer les statistiques de conversion du schéma de base de données de la liste des objets de stockage de base de données.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Amélioration de la conversion de l'UPDATE instruction sans la FROM clause.
Teradata	Amazon Red	Support implémenté du type de VARBYTE données dans le code converti.
Teradata BTEQ	AWS Glue	Résolution d'un problème en raison duquel l' AWS Glue option Convertir en était désactivée dans le menu contextuel.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'un problème en raison duquel des types de données manquaient dans le code converti.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'un problème en raison duquel les variables de substitution étaient incorrectement citées dans le code converti.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'un problème de conversion des variables de substitution avec des valeurs dans les FastLoad scripts.
Vertica	Amazon Red	Support implémenté du type de TIME données dans le code converti.
Vertica	Amazon Red	Conversion améliorée des ORDER BY expressions SELECT DISTINCT et des expressions.
Vertica	Amazon Red	Ajout du support pour la conversion des contraintes.
Vertica	Amazon Red	Résolution d'une erreur en raison de laquelle un rapport d'évaluation n'était pas enregistré sous forme de fichier CSV (valeurs séparées par des virgules).

## Notes de mise à jour pour la AWS SCT version 659

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Amélioration du nouvel assistant de projet qui génère un rapport d'évaluation combiné pour plusieurs bases de données sources.
Tous	Tous	Correction d'un problème en raison duquel le pack d'extension n'était pas créé dans des projets incluant plusieurs bases de données source et cible.
Tous	Tous	Conversion améliorée du code SQL intégré au code source de l'application.
Tous	Tous	Ajout de la possibilité d'exécuter des scripts à partir de différents dossiers dans l'interface de AWS SCT ligne de commande.
Tous	Amazon Redshift	Le message d'avertissement envoyé lorsque les utilisateurs choisissent Exécuter l'optimisation dans les projets de migration avec la plateforme de base de données cible virtuelle Amazon Redshift a été amélioré.
Tous	Aurora PostgreSQL	Mise en œuvre du support de la version majeure 13 de PostgreSQL sur Aurora PostgreSQL Compatible Edition en tant que cible de migration.
Tous	Amazon RDS for MySQL	Implémentation par défaut de la conversion de code insensible aux majuscules et minuscules.
Analyses Azure Synapse	Amazon Redshift	Résolution d'une erreur lors de laquelle la connexion à une base de données source échouait dans l'interface de ligne de commande.
Microsoft SQL Server	PostgreSQL	Conversion améliorée des procédures qui incluent des UPDATE instructions avec des conditions de jointure.

Source	Cible	Nouveautés, améliorations ou correctifs
	Aurora PostgreSQL L	
Microsoft SQL Server	PostgreSQL L  Aurora PostgreSQL L	Conversion améliorée des déclencheurs, des procédures stockées et des fonctions qui incluent la valeur après le signe égal.
Microsoft SQL Server	PostgreSQL L  Aurora PostgreSQL L	Correction d'une erreur de transformateur pour les procédures avec l'DELETE instruction et l'ORopérateur.
Microsoft SQL Server	PostgreSQL L  Aurora PostgreSQL L	Conversion améliorée de la OUTPUT clause.
Microsoft SQL Server DW	Amazon Redshift et AWS Glue	Conversion améliorée du type NUMERIC de données.
Microsoft SQL Server DW	Amazon Red	Conversion améliorée des vues dont l'alias de table porte le même nom que la table d'origine.
Microsoft SSIS	AWS Glue	Correction d'un problème en raison duquel les informations d'identification de AWS Glue connexion n'étaient pas affichées dans la fenêtre de configuration des connexions.

Source	Cible	Nouveautés, améliorations ou correctifs
Netezza	Amazon Red	Ajout de la possibilité de répéter chaque jour l'exécution des tâches de migration des données de capture des données modifiées (CDC).
Netezza	Amazon Red	Correction d'un problème en raison duquel l'onglet Tâches devenait inactif après avoir annulé l'enregistrement d'un agent d'extraction de données.
Netezza	Amazon Red	Correction d'un problème en raison duquel la confirmation de l'enregistrement de l'agent de migration de données ne s'affichait pas dans l'interface utilisateur.
Netezza	Amazon Red	Correction d'un problème en raison duquel une connexion à une base de données source échouait en raison d'une erreur du chargeur.
Netezza	Amazon Red	Résolution d'une erreur qui empêchait les agents de migration de données de s'exécuter après l'ouverture d'un projet enregistré.
Oracle	Amazon RDS for Oracle	Support implémenté d'Oracle Unified Auditing.
Oracle	PostgreSQL L  Aurora PostgreSQL L	Conversion de code SQL implémentée dans les applications C#. Pour de plus amples informations, veuillez consulter <a href="#">Code SQL dans les applications C#</a> .
Oracle	PostgreSQL L  Aurora PostgreSQL L	Mise en œuvre d'une nouvelle logique de transformation pour les noms d'objets distinguant majuscules et minuscules afin d'améliorer la visibilité des modifications de conversion du code. AWS SCT convertit les noms d'objets en majuscules en minuscules. L'inverse est également vrai ; AWS SCT convertit les noms d'objets en minuscules en majuscules. Les autres noms d'objets et mots réservés sont convertis sans modification.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL Aurora PostgreSQL	Conversion améliorée des partitions de hachage sans NOT NULL contrainte.
Oracle	Aurora PostgreSQL	Ajout du support pour la conversion d'Oracle CHECKFOREIGN KEY, et NOT NULL des contraintes avec la ENABLE NOVALIDATE clause.
Oracle DW	Amazon Redshift	Correction d'un problème en raison duquel les valeurs incorrectes pour les nombres à virgule flottante étaient migrées.
Oracle DW	Amazon Redshift et AWS Glue	Résolution d'un problème lié aux colonnes vides du rapport d'évaluation de la migration de la base de données dans un fichier CSV (valeurs séparées par des virgules).
SAP ASE	PostgreSQL Aurora PostgreSQL	Correction d'un problème lié à une interruption inattendue de la conversion.
Snowflake	Amazon Redshift	Conversion améliorée du type VARIANT de données.
Teradata	Amazon Redshift	Conversion améliorée de la COLLECT STATISTICS déclaration.
Teradata	Amazon Redshift	Correction d'un problème en raison duquel l'action 9998 n'apparaissait pas lors de la conversion de vues imbriquées en colonnes. PERIOD
Teradata	Amazon Redshift et AWS Glue	Correction d'un problème en raison duquel une plate-forme AWS Glue cible virtuelle ne s'affichait pas dans l'interface utilisateur après l'ouverture d'un projet enregistré.



Source	Cible	Nouveautés, améliorations ou correctifs
Teradata BTEQ	AWS Glue	Correction d'un problème en raison duquel la conversion vers une plate-forme AWS Glue cible virtuelle n'était pas prise en charge après l'ouverture d'un projet enregistré.
Teradata BTEQ	Amazon Redshift RSQL	Amélioration de la mise en évidence de la syntaxe du code converti.
Teradata BTEQ	Amazon Redshift RSQL	Mise en œuvre de la vérification des valeurs des paramètres après le téléchargement. Les valeurs non prises en charge sont mises en évidence dans l'onglet Variables.
Vertica	Amazon Red	Conversion implémentée des fonctions d'agrégation.
Vertica	Amazon Red	Implémentation de la conversion des projections en vues matérialisées et amélioration de l'interface utilisateur qui affiche le code source des projections.

## Notes de mise à jour pour la AWS SCT version 658

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Intégration fournie avec AWS Secrets Manager. Vous pouvez désormais utiliser les informations de connexion à la base de données stockées dans Secrets Manager.
Tous	Tous	Ajout de la prise en charge des scripts au format YAML dans l'interface de AWS SCT ligne de commande.
Tous	Amazon Red	Support implémenté des points de terminaison d'interface Amazon S3 (VPCE) dans les agents d'extraction de données.
Tous	Amazon Red	Ajout de la prise en charge de la plate-forme de base de données cible virtuelle Amazon Redshift en plus de la solution Amazon AWS Glue Redshift déjà prise en charge et de sa combinaison.

Source	Cible	Nouveautés, améliorations ou correctifs
Greenplum	Amazon Red	Correction d'un problème en raison duquel l'option Enregistrer en tant que SQL n'enregistrait pas le code SQL converti dans un fichier.
IBM Db2 LUW	Aurora MySQL	Conversion améliorée pour prendre en charge les nouvelles fonctionnalités de l'édition compatible Amazon Aurora avec MySQL 8.0.
Base de données Microsoft Azure SQL		
Microsoft SQL Server		
Oracle		
SAP ASE		
Microsoft SQL Server	Aurora MySQL	Correction d'un problème en raison duquel l'action 810 n'apparaissait pas lorsque cela était nécessaire.
	Aurora PostgreSQL	
	MySQL	
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Conversion améliorée des procédures avec UPDATEDELETE, et INSERT instructions.
	PostgreSQL	

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Correction d'un problème en raison duquel l'action 7810 n'apparaissait pas lorsque cela était nécessaire.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée d'une EXEC instruction imbriquée dans une IF...ELSE instruction.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des vues indexées.
Netezza	Amazon Redshift	Agents de migration de données améliorés en suivant les transactions en temps réel pendant le chargement complet dans le cadre de l'opération de capture des données de modification (CDC). Vous pouvez désormais arrêter les tâches de migration de données si le démarrage de la session CDC est prévu à une certaine heure. Vous pouvez également voir le niveau de journalisation des erreurs dans la console après avoir arrêté une tâche avec CDC.
Oracle	Tous	Amélioration du chargeur de tables pour garantir le AWS SCT chargement des objets avec des options de partage.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée de la SYSDATE fonction et ajout de la possibilité de modifier le fuseau horaire dans les paramètres de conversion.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Résolution d'un problème en raison duquel les instructions dynamiques n'étaient pas converties.
Oracle	Aurora PostgreSQL L  PostgreSQL L	Correction d'un problème en raison duquel le code converti n'incluait pas les noms générés par le système.
Oracle Oracle DW	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des SELECT instructions imbriquées dans les déclencheurs.
Oracle DW	Amazon Red	Conversion améliorée des TO_TIMESTAMP_TZ fonctions TO_DATE, TO_TIMESTAMP , et dans le pack d'extension.
Snowflake	Amazon Red	Ajout d'une option permettant d'enregistrer le code SQL converti dans différents fichiers pour chaque objet ou pour chaque instruction.
Teradata	Amazon Red	Conversion améliorée de la CONCAT fonction.
Teradata	Amazon Red	Conversion améliorée d'une SELECT instruction imbriquée dans une WHERE clause.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Résolution d'un problème de conversion incorrecte SET des MULTISSET tables après que les utilisateurs aient supprimé et recréé une table.
Teradata	Amazon Red	Conversion améliorée des procédures qui incluent une WITH clause.
Teradata	Amazon Red	Conversion améliorée du type DATE de données.
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel une erreur de transformation inattendue s'est produite lors de la conversion FastExport des scripts.
Teradata BTEQ	Amazon Redshift RSQL	Ajout du support pour la conversion d'un index de jointure en vue matérialisée.
Teradata BTEQ	Amazon Redshift RSQL	Ajout de la prise en charge de la conversion d'une TITLE définition comprenant plusieurs lignes.
Teradata BTEQ	Amazon Redshift RSQL	Résolution d'un problème en raison duquel la taille d'un type de données géospatiales n'était pas convertie.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'un problème en raison duquel les noms des paramètres étaient convertis en minuscules.
Teradata BTEQ	Amazon Redshift RSQL	Correction d'un problème lorsqu'une procédure stockée imbriquée dans une MACRO instruction n'était pas convertie.
Vertica	Amazon Red	Conversion améliorée de l'ALLopérateur.
Vertica	Amazon Red	Résolution d'un problème en raison duquel l'Use Union all view?option des paramètres de conversion n'était pas appliquée.

Source	Cible	Nouveautés, améliorations ou correctifs
Vertica	Amazon Red	Conversion améliorée des types de TIME WITH TIMEZONE données TIME et.
Vertica	Amazon Red	Résolution d'un problème lié au chargement des tables flexibles.

Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 657

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Mise à niveau d'Apache Log4j vers la version 2.17 pour résoudre les problèmes de sécurité.
Tous	Amazon Red	Projets d'optimisation du schéma améliorés, dans lesquels les statistiques de gestion clés n'étaient pas enregistrées dans le AWS SCT projet.
Amazon Red	Amazon Red	Correction d'un problème lié à la mise à jour des informations du serveur.
Apache Cass	Amazon DynamoDB	Correction d'un problème lié aux règles de mappage lors de l'utilisation de l'interface de AWS SCT ligne de commande.
Apache Cass	Amazon DynamoDB	Résolution d'un problème lorsque la tâche de migration n'était pas créée en raison d'un titre mis à jour dans le certificat.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Correction d'un problème qui empêchait l'action 7672 d'apparaître lors de la conversion des procédures Microsoft SQL Server en SQL dynamique.

Source	Cible	Nouveautés, améliorations ou correctifs
Base de données Azure SQL	Aurora PostgreSQL L	Conversion améliorée des fonctions à valeur tabulaire.
Microsoft SQL Server	PostgreSQL L	
Base de données Azure SQL	Aurora PostgreSQL L	Résolution d'un problème en raison duquel l'OUT argument d'une procédure stockée avec la valeur de retour par défaut n'était pas converti en INOUT argument.
Microsoft SQL Server	PostgreSQL L	
Greenplum	Amazon Red	Stratégies d'optimisation améliorées en repérant les tables et les colonnes les plus utilisées dans la QueryLog table.
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Correction de problèmes liés à la conversion des éléments suivants : <ul style="list-style-type: none"> <li>• Opérateur d'affectation par concaténation de chaînes () +=</li> <li>• Fonction SCOPE_IDENTITY</li> <li>• Type de données varchar(max)</li> </ul>
Microsoft SQL Server	Aurora PostgreSQL L  PostgreSQL L	Conversion améliorée des vues avec des fonctions non prises en charge.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Correction d'un problème en raison duquel les fonctions non prises en charge en tant qu'argument d'une autre fonction étaient incorrectement converties.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Conversion améliorée des références aux tables de transition.
Microsoft SQL Server DW	Amazon Redshift	La catégorie des fonctions d'agrégation a été ajoutée à l'arborescence des métadonnées de la base de données source.
Microsoft SQL Server DW	Amazon Redshift	Conversion améliorée du type TIME de données.
Analyses Azure Synapse Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	Correction d'un problème en raison duquel DROP les CREATE scripts n'étaient pas enregistrés lors de l'utilisation d'une plate-forme de base de données cible virtuelle.



Source	Cible	Nouveautés, améliorations ou correctifs
Services d'intégration Microsoft SQL Server	AWS Glue	Problème résolu : les scripts des objets source ne s'affichaient pas dans l'interface utilisateur.
Netezza	Amazon Red	Stratégies d'optimisation améliorées en choisissant le tableau des faits et les dimensions appropriées pour la collocation.
Oracle	Aurora PostgreSQL PostgreSQL	Résolution d'un problème de conversion correcte des déclencheurs Oracle, qui utilisent des numéros de séquence.
Oracle	Aurora PostgreSQL PostgreSQL	Conversion améliorée des vues grâce à des liens vers des bases de données publiques.
Oracle DW	Amazon Red	Stratégies d'optimisation améliorées en analysant la cardinalité des colonnes d'index.
Oracle DW	Amazon Red	Correction d'un problème en raison duquel les fonctions scalaires personnalisées définies par l'utilisateur avec concaténation de chaînes étaient mal converties.
Snowflake	Amazon Red	Correction d'un problème en raison duquel l'option Enregistrer en tant que SQL ne s'affichait pas dans l'interface utilisateur.
Teradata	Amazon Red	Correction d'un problème en raison duquel la collecte de statistiques échouait, <code>LOADER ERROR</code> sauf exception.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Correction d'un problème en raison duquel l'option Créer un rapport ne s'affichait pas dans l'interface utilisateur.
Teradata	Amazon Red	Conversion améliorée de la CAST fonction.
Teradata	Amazon Red	Correction d'une conversion interrompue pour <code>ST_Line_Interpolate_Point</code> .
Teradata	Amazon Red	Suppression d'une valeur inattendue dans le chemin de la bibliothèque Python.
Teradata	Amazon Redshift RSQL	Correction d'une erreur de résolution qui apparaissait lors de la conversion de plusieurs FastLoad scripts.
Teradata BTEQ	Amazon Redshift RSQL	Conversion améliorée des types de données de DATABASE commande et de géométrie.
Teradata BTEQ	AWS Glue	Correction d'un problème lié à une synchronisation incorrecte des scripts source et cible dans l'interface utilisateur.

Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 656

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la prise en charge de plusieurs bases de données sources et cibles au sein d'un même projet. Les utilisateurs peuvent désormais créer des règles de mappage correspondant aux différents schémas de base de données et aux différentes plateformes cibles d'un même projet.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout de la prise en charge des plateformes de base de données cibles virtuelles. Les utilisateurs n'ont désormais plus besoin de se connecter à une base de données cible pour voir comment AWS SCT convertir le schéma de leur base de données source.
Tous	Tous	Améliorations de l'interface : <ul style="list-style-type: none"> <li>• Les options Connect to the server et Disconnect from the server ont été ajoutées aux arborescences de métadonnées source et cible.</li> <li>• Ajout d'une option permettant de supprimer un serveur de base de données du AWS SCT projet.</li> </ul>
Cassandra	Amazon DynamoDB	Résolution d'un problème de recherche en raison duquel la CASSANDRA_HOME variable n'incluait pas de barre oblique (/) après <code>cassandra.yaml</code> le conf dossier.
Cassandra	Amazon DynamoDB	Ajout du support de l'Amazon Machine Image (AMI) pour Amazon Linux 2.
Cassandra	Amazon DynamoDB	Message d'erreur amélioré fourni lorsqu'une clé incorrecte est donnée pour Cassandra.
Cassandra	Amazon DynamoDB	Conversion améliorée en modifiant une propriété du <code>cassandra-env.yaml</code> fichier en fonction de la version de la base de données cible.
Cassandra	Amazon DynamoDB	La version Java du centre de données Cassandra cible a été augmentée à 1.8.0.
Greenplum	Amazon Red	Stratégies d'optimisation améliorées dans les paramètres du projet.

Source	Cible	Nouveautés, améliorations ou correctifs
Greenplum	Amazon Red	Résolution d'un problème de migration de données en raison duquel les objets n'étaient pas appliqués à la base de données avec cette erreur :An I/O error occurred while sending to the backend .
Greenplum Microsoft SQL Server DW	Amazon Red	Résolution d'un problème en raison duquel l'Apply RTRIM to string column option ne s'affichait pas dans l'interface utilisateur.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Ajout de la prise en charge de Babelfish pour Aurora PostgreSQL en tant que plate-forme cible. Les utilisateurs peuvent désormais créer un rapport d'évaluation pour estimer la migration de SQL Server vers Babelfish pour Aurora PostgreSQL.
Netezza	Amazon Red	Stratégies d'optimisation améliorées dans les paramètres du projet.
SAP ASE	Aurora PostgreSQL  PostgreSQL	Implémentation de la possibilité de générer des noms uniques pour les index.
SAP ASE	Aurora PostgreSQL  PostgreSQL	Correction d'un problème lié à une colonne d'index dupliquée dans le script cible.
Snowflake	Amazon Red	Résolution d'un problème en raison duquel les options Masquer les schémas vides, Masquer les bases de données vides et Masquer les bases de données/schémas système n'étaient pas affichées dans l'interface utilisateur.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift RSQL	Ajout de la prise en charge de la conversion des scripts de MultiLoad travail Teradata en scripts Amazon Redshift RSQL.
Teradata	Amazon Redshift RSQL	Correction d'un problème lié à la conversion des variables de substitution dans FastLoad les FastExport scripts et.
Teradata	Amazon Redshift RSQL	Correction d'un problème en raison duquel les actions ne s'affichaient pas dans l'onglet Actions après le passage de l'onglet Résumé.
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel une erreur se produisait après la génération du rapport lors de la conversion FastExport des scripts.
Teradata	Amazon Redshift RSQL	Problèmes de formatage résolus après la conversion des scripts shell.
Teradata	Amazon Redshift RSQL	Correction d'un problème qui faisait que AI 13177 était désormais commenté dans un script converti.
Teradata	Amazon Red	Correction d'une erreur de conversion des tables temporelles.
Teradata	Amazon Red	Conversion améliorée de la SET QUERY_BAND déclaration.
Teradata	Amazon Red	Correction d'une erreur de conversion de l'NORMALIZE opération.
Vertica	Amazon Red	Amélioration de la description de l'AI 17008.

## Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 655

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift RSQL	Correction d'un problème qui faisait en sorte que tous les problèmes d'évaluation apparaissent dans les rapports en cas FastLoad d'utilisation ou MultiLoad d'utilisation.
Teradata	Amazon Redshift RSQL	Ajout de la prise en charge de la conversion des scripts de FastExport travail Teradata en scripts Amazon Redshift RSQL.
Teradata	Amazon Redshift RSQL	Correction d'un problème qui empêchait l'action Enregistrer le manifeste dans S3 d'être activée en mode hors ligne lors de l'utilisation FastLoad.
Teradata	Amazon Redshift RSQL	Correction d'un problème garantissant l'application des règles de mappage pour les scripts tels que FastLoad.
Greenplum	Amazon Red	La version minimale du pilote prise en charge pour Greenplum a été augmentée à 42.2.5.
Greenplum	Amazon Red	Ajout d'une connexion à Greenplum via SSL avec la version du pilote 42.2.5 ou supérieure.
Oracle DW	Amazon Red	Support amélioré pour l'exécution de fonctions scalaires personnalisées définies par l'utilisateur (UDF) dans un autre UDF.
Oracle DW	Amazon Red	Correction d'un problème en raison duquel les fonctions n'étaient pas appliquées à la base de données avec cette erreur : <code>Failed to compile udf</code>
Oracle DW	Amazon Red	Conversion améliorée grâce à l'utilisation de déclarations de type appropriées, telles que, <code>pls-type</code> pour les paramètres <code>%ROWTYPE</code> .

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift RSQL	Résolution d'un problème en raison duquel les problèmes d'évaluation des types d'informations n'apparaissaient pas dans le rapport.
Teradata	Amazon Redshift RSQL	Résolution d'une erreur de transformation après la conversion de certains scripts.
Teradata	Amazon Redshift RSQL	Correction d'un problème qui faisait qu'un problème était désormais commenté dans un script converti.
Teradata	Amazon Red	Résolution d'un problème où FastExport ->EXPORT -> « null » s'affichait à la place de « CAST » après la conversion.
Teradata	Amazon Red	Résolution d'un problème en raison duquel certaines fonctions d'un pack d'extension échouaient lorsqu'elles étaient appliquées avec Cause:[JDBC Driver]String index out of range: 0 la version 1.2.43 du pilote
Teradata	Amazon Red	CONVERSION DE TABLE SET : l'émulation de table SET a été ajoutée pour les instructions d'insertion-sélection.
Teradata	Amazon Red	CAST : prend en charge le casting de types de données supplémentaires.
Teradata	Amazon Red	Correction d'une erreur de conversion pour « other_current_time_01 »
Teradata	Amazon Red	Teradata FastExport — Amazon Redshift RSQL : conversion améliorée des commandes Teradata — Field FastExport
Teradata	Amazon Red	Teradata FastExport — Amazon Redshift RSQL : conversion améliorée des commandes Teradata — Layout FastExport

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL Aurora PostgreSQL	Résolution d'un problème en raison duquel le script cible des objets avec l'instruction <code>SAVE EXCEPTIONS</code> était modifié après la reconversion.
Oracle	PostgreSQL Aurora PostgreSQL	Résolution d'un problème en raison duquel un champ erroné était spécifié dans la clause <code>ORDER BY</code> après <code>proc_cursor_with_calc_columns</code> la conversion.
Oracle	PostgreSQL Aurora PostgreSQL	Résolu : une déclaration de <code>aws_oracle_ext\$array_id\$temporary</code> variable supplémentaire est requise dans une conversion <code>ASSOCIATIVE COLLECTION</code> .
Oracle	PostgreSQL Aurora PostgreSQL	Résolu : mauvaise conversion d'une CLÉ PRIMAIRE portant le même nom qu'un INDEX appartenant à la même table.

#### Problèmes résolus :

- Améliorations générales



## Notes de mise à jour pour la AWS SCT version 654

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL L  Aurora PostgreSQL L	Résolution d'un problème lié aux pseudocolonnes de requête hiérarchique et à l'erreur d'analyse des colonnes PRIOR.
Oracle	PostgreSQL L  Aurora PostgreSQL L	Résolution d'un problème de conversion correcte d'un commentaire multiligne contenant une barre oblique et un astérisque (/*) .
Oracle	PostgreSQL L  Aurora PostgreSQL L	Ajout de l'émulation USER_COL_COMMENTS de la vue système au pack d'extension.
Oracle	PostgreSQL L  Aurora PostgreSQL L	Conversion améliorée des littéraux entre guillemets.
DB2 LUW	PostgreSQL L  Aurora PostgreSQL L	Conversion améliorée des instructions LABEL qui ajoutent ou remplacent des étiquettes dans les descriptions des tables, des vues, des alias ou des colonnes.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Aucun	La table système SYS.USER\$ a été remplacée par la vue DBA_USERS et les requêtes ont été améliorées.
Oracle DW	Amazon Red	Requêtes de métadonnées Oracle DW mises à jour.
Teradata	Amazon Redshift RSQL	Ajout de la prise en charge de la conversion des scripts shell, Teradata FastLoad et Teradata Basic Teradata Query (BTEQ) en scripts RSQL Amazon Redshift.
Teradata BTEQ	Amazon Redshift RSQL	Le problème de conversion incorrecte de « merge_01 » a été résolu.
Teradata BTEQ	Amazon Redshift RSQL	Résolution du problème selon lequel End or Identify (EOI) apparaissait à la fin d'un script sur une nouvelle ligne.
Azure Synap	Amazon Red	Message d'erreur amélioré fourni en cas de mot de passe incorrect pour Azure Synapse.
Teradata	Amazon Red	Conversion améliorée de l'instruction UPDATE pour transférer le bon nom d'alias conformément à la norme Teradata.
Teradata	Amazon Red	Résolution d'une erreur de conversion du curseur à cause de laquelle aucune action n'était reçue.
Teradata	Amazon Red	Résolution d'un problème de suppression de lignes lors d'une conversion TD_NORMALIZE_OVERLAP.
Teradata	Amazon Red	Supporte désormais une vérification stricte de la date pour la fonction améliorée TO_DATE.
Teradata	Amazon Red	Conversion améliorée de la fonction intégrée TO_NUMBER (n).
Teradata	Amazon Red	Résolution d'un problème en raison duquel la catégorie Schémas était absente de l'arborescence des métadonnées.

Source	Cible	Nouveautés, améliorations ou correctifs
Greenplum	Amazon Red	Ajout de la sélection GP_SEGMENT_ID à la liste lors de la création d'une partition virtuelle pour une table Greenplum.
Greenplum	Amazon Red	Résolution d'un problème en raison duquel les fonctions n'étaient pas appliquées sur la cible.
MS SQL Server DW	Amazon Red	Résolution d'un problème où une erreur de transformation se produisait après une conversion sans AI 9996.
MS SQL Server DW	Amazon Red	Résolution d'un problème en raison duquel une erreur était enregistrée lors de l'ouverture de l'assistant du pack d'extension.
MS SQL Server DW	Amazon Red	Résolution d'un problème lié à l'utilisation d'un style de commentaire incorrect pour les fonctions Python de Redshift.
Netezza	Amazon Red	Résolution d'un problème en raison duquel un pack d'extension Netezza—Redshift avec un profil n'a pas pu être créé. AWS
Teradata	Amazon Redshift RSQL	Conversion améliorée de la commande FastLoad SESSIONS.
Teradata	Amazon Redshift RSQL	Rapports d'évaluation des FastLoad scripts améliorés.
Teradata	Amazon Redshift RSQL	L'action FastLoad WRITER Save to S3 a été implémentée.
Teradata	Amazon Redshift RSQL	Problème résolu : les boutons FastLoad Enregistrer le script \ Enregistrer le manifeste dans s3 n'étaient pas actifs.
Teradata	Amazon Redshift RSQL	Résolution d'un problème selon lequel FastLoad multifile_script ne créait qu'un seul fichier manifeste après la conversion au lieu des trois fichiers attendus.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Redshift RSQL	Résolution d'un problème FastLoad d'affichage de dossiers supplémentaires dans un chemin S3.
Teradata	Amazon Redshift RSQL	Résolution d'un problème FastLoad lié au nom incorrect du fichier manifeste dans un chemin S3.

Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 653

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL Aurora PostgreSQL	Implémentation de la capacité de convertir le SQL dynamique créé dans des fonctions ou des procédures appelées.
Oracle	PostgreSQL Aurora PostgreSQL	Conversion SQL dynamique améliorée : paramètres intégrés sous forme de variables de liaison.
Oracle DW 18, 19	Amazon Redshift	Améliorations apportées à la conversion d'Oracle vers Redshift : intégration améliorée de la conversion. LISTAGG agrégé ; LISTAGG analytique.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle DW 18,19	Amazon Red	Améliorations apportées à la conversion d'Oracle vers Redshift : recherchez de nouvelles fonctionnalités.
Vertica	Amazon Red	Améliorations apportées à la conversion de Vertica vers Redshift : connexion SSL vers JDBC avec SSL=true.
MS SQL Server DW	Amazon Red	Améliorations apportées à la conversion de MS SQL Server vers Redshift : tables externes.
Teradata	Amazon Red	Améliorations de la conversion de Teradata vers Redshift : opération s arithmétiques sur les types de données INTERVAL.
Teradata	Amazon Red	Améliorations de la conversion de Teradata vers Redshift : Support pour les alias de colonnes latérales.
Oracle	Aucun	<p>Les requêtes Loader suivantes sont désormais utilisées à la DBA_USERS place de SYS.USER\$ :</p> <ul style="list-style-type: none"> <li>• get-tree-path-list- by-name-path .sql</li> <li>• estimate-table-or-view- constraints-by-schema .sql</li> <li>• estimate-table-or-view- constraints-by-selected-schemas .sql</li> </ul>
Teradata	Amazon Red	Amélioration de l'alignement des commentaires lorsque SCT convertit les macros Teradata en procédures stockées Redshift.
Oracle DW	Amazon Red	Conversion améliorée des éléments du format Date/Timestamp :, et TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ
Teradata	Amazon Red	Erreur de conversion du curseur Teradata résolue.
Teradata	Amazon Red	Le problème à l'origine de la suppression des attributs TD_NORMALIZE_OVERLAP de lors de la conversion a été résolu.
Teradata	Amazon Red	Résolution d'un problème en raison duquel MAX la fonction était ignorée lors de la conversion d'une requête par SCT.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	SCT convertit désormais la fonction Teradata CHARACTERS en fonction Redshift LENGTH.
Teradata	Amazon Red	SCT prend désormais en charge la conversion de FORMAT en TO_CHAR pour les formats les plus couramment utilisés.
Tous	Tous	Conversion améliorée des routines cryptées.

### Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 652

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	PostgreSQL	Verrouillage <code>sp_getapplock</code> et <code>sp_releaseapplock</code> fonctions de verrouillage des applications ajoutés.
Aucun	Amazon Red	Amélioration de l'interface de ligne de commande (CLI) : mise en œuvre du mode Script Command.
Oracle	PostgreSQL Aurora PostgreSQL	Échantillonnage de paramètres de routine implémenté dans le SQL dynamique.
Oracle	PostgreSQL	Améliorations de la conversion en SQL dynamique créé dans des fonctions ou des procédures appelées.

Source	Cible	Nouveautés, améliorations ou correctifs
	Aurora PostgreSQL	
Microsoft SQL Server  Oracle  DB2 LUW	Aurora PostgreSQL	Chaque fonction lambda n'est déployée et configurée qu'une seule fois via une politique, et les fonctions lambda communes sont réutilisées pour toutes les sources possibles.
DB2 LUW	PostgreSQL	Problème résolu à l'origine du message d'erreur « 9996 — Sévérité critique — Une erreur de transformateur s'est produite » lors de l'utilisation de DB2 LUW comme source.
Teradata	Amazon Redshift	Support des expressions de table récursives lors du prochain lancement d'Amazon Redshift.
Azure Synapse	Amazon Redshift	Règles d'optimisation du schéma mises en œuvre.
Teradata	Amazon Redshift	Support de conversion des fuseaux horaires des macros Teradata en procédures stockées Redshift.
Teradata	Amazon Redshift	Support arithmétique sur les valeurs PERIOD.
Teradata	Amazon Redshift	Support de conversion des expressions de table communes récursives Teradata (RECURSIVE CTE).
Teradata	Amazon Redshift	Support des identifiants sensibles aux majuscules et minuscules via les paramètres utilisateur, <code>enable_case_sensitive_identifier</code> . Ainsi, « COLUMN_NAME » et « Column_Name » deviennent des noms de colonne différents.
Teradata	Amazon Redshift	Le problème de type de données décimal a été résolu afin que les champs décimaux soient convertis avec la même précision.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Résolution d'un problème lié à la conversion arithmétique par intervalles afin que la soustraction arithmétique par intervalles soit correctement convertie.
Teradata	Amazon Red	Conversion de type TERADATA NUMBER to DATE améliorée.
Teradata	Amazon Red	Conversion améliorée des types Teradata DATE vers NUMBER
Teradata BTEQ	Amazon Red	Conversion améliorée des types de données PERIOD.
Teradata	Amazon Red	Le problème lié au chargement des métadonnées d'une table contenant des colonnes GEOMETRY a été résolu, de sorte qu'elle se charge désormais correctement depuis Teradata.
Teradata	Amazon Red	Support de conversion des instructions de fusion lors de la conversion de macros Teradata en procédures stockées Redshift.
Teradata	Amazon Red	Conversion améliorée de macros simples lors de la migration de Teradata vers Redshift.
Teradata	Amazon Red	Garantie que la conversion des instructions Teradata UPDATE reporte le nom d'alias correct conformément à la norme Teradata.

Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 651

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	AWS SCT Rapports améliorés pour mettre à jour les liens vers les actions de conversion recommandées répertoriées.



Source	Cible	Nouveautés, améliorations ou correctifs
Serveur MS SQL	PostgreSQL	Ajout du support pour la conversion de STR( ) fonction.
Serveur MS SQL	PostgreSQL	Ajout de la prise en charge de la conversion de l'opérateur EXOR au niveau du bit (dans ^ Microsoft SQL Server) en PostgreSQL en tant qu'opérateur. #
Oracle	PostgreSQL	Résolution d'un problème de blocage de la aws_oracle_ext.UNISTR(null) fonction du pack d' AWS SCT extension NULL sur une cible PostgreSQL. AWS SCT gère désormais leNULL.
Teradata BTEQ	Amazon Redshift RSQL	Améliorations apportées à la conversion pour résoudre un problème selon lequel la conversion d'Amazon Redshift RSQL MERGE provoquait une erreur de transformation.
Oracle DW	Amazon Red	Implémentation d'éléments intégrés améliorés.
Oracle DW	Amazon Red	Ajout d'améliorations axées sur les fonctionnalités de métadonnées, notamment le partitionnement automatique des listes (TBL_PART_LIST_AUTO), les listes à colonnes multiples (TBL_PART_MULTILIST) et les références d'intervalles (TBL_PART_RANGE_INTVL_REF).
none	Amazon Red	Augmentation des limites des tables de partitions pour les partitions physiques utilisées pour UNION ALL les conversions.
Teradata	Amazon Red	Améliorations apportées à la conversion du champ d'application des rapports d'évaluation.
Teradata	Amazon Red	Améliorations apportées aux conversions Teradata MACRO complexes.
Teradata	Amazon Red	Conversion améliorée des macros Teradata en procédures stockées Amazon Redshift tout en commentant les instructions SQL non prises en charge.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Résolution d'un problème selon lequel la conversion de macros Teradata en procédures stockées Amazon Redshift entraînait des références de nom d'alias incorrectes.
Teradata	Amazon Red	Conversion améliorée de l'QUALIFY instruction Teradata.
Teradata	Amazon Red	Conversion améliorée pour transférer les commentaires vers Amazon Redshift et conserver un historique des modifications effectuées sur la vue.
Teradata	Amazon Red	Résolution d'un problème en raison duquel la clause RESET WHEN n'entraînait pas la conversion correcte.
Teradata BTEQ	Amazon Red	Conversion améliorée des scripts BTEQ contenant des instructions MERGE.
Teradata	Amazon Red	Ajout de fonctions intégrées pour améliorer la conversion des champs de type de données PERIOD.
Microsoft SQL Server	Amazon Red	Cartographie améliorée du type de données de transformation pour le type de données TIME.
Tous	Tous	Accès ajouté à la publication initiale du manuel de référence de la AWS Schema Conversion Tool CLI au format PDF. Voir la <a href="#">référence de la AWS Schema Conversion Tool CLI</a> .

## Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour AWS SCT Build 650

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	<p>Utilisation actualisée et améliorée des agents d'extraction, notamment :</p> <ul style="list-style-type: none"> <li>• Configuration à utiliser avec un stockage partagé et un agent de copie dédié.</li> <li>• Exportation et importation de tâches d'extraction de données d'un projet à l'autre.</li> <li>• Support pour Azure SQL Data Warehouse (Azure Synapse) en tant que source.</li> <li>• Utilisation du partitionnement Netezza natif.</li> </ul> <p>Pour de plus amples informations, veuillez consulter <a href="#">Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool</a>.</p>
Tous	Amazon RDS PostgreSQL 13	AWS SCT prend désormais en charge Amazon RDS PostgreSQL 13 comme cible.
Microsoft SQL Server	Aurora PostgreSQL	Conversion améliorée d'un jeu de résultats d'une procédure Microsoft SQL Server vers une cible Aurora PostgreSQL.
Oracle DW	Amazon Redshift	Améliorations apportées à la conversion d'Oracle vers Amazon Redshift.
Oracle DW	Amazon Redshift	Améliorations apportées à la conversion des instructions SQL dynamiques.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle DW	Amazon Red	Améliorations apportées à la conversion SQL UDF.
Oracle DW	Amazon Red	Message clarifié qui AWS SCT ne prend pas en charge la conversion de TABLES EXTERNES.
Oracle DW	Amazon Red	Fonctions de conversion intégrées améliorées.
Teradata BTEQ	Amazon Redshift RSQL	Amélioration de la gestion des paramètres de substitution dans les scripts BTEQ lors de l'utilisation AWS SCT de l'interface graphique.
Microsoft SQL Server DW	Tous	Mise à niveau de la version minimale du pilote JDBC prise en charge pour Microsoft SQL Server, Azure et Azure Synapse.
Microsoft SQL Server		
Azure		
Azure Synap		

#### Problèmes résolus :

- Teradata : Améliorations supplémentaires relatives à la conversion de macros [RÉSOLU]
- Des caractères spéciaux se sont échappés dans la cible, provoquant des erreurs SQL et une refonte est nécessaire pour les rétablir [RÉSOLU]
- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 649

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server DW	Amazon Red	Améliorations de la conversion de MSSQL vers Amazon Redshift pour prendre en charge les tables temporelles.
Oracle DW	Amazon Red	Améliorations des fonctions intégrées mises en œuvre, telles que :  Fonctions de conversion <ul style="list-style-type: none"><li>• TO_BINARY_DOUBLE</li><li>• TO_BINARY_FLOAT</li><li>• TO_NUMBER</li><li>• TO_DATE</li><li>• TO_TIMESTAMP</li><li>• TO_TIMESTAMP_TZ</li><li>• TO_DSINTERVAL</li><li>• TO_YMINTERVAL</li><li>• VALIDE_CONVERSION</li></ul>
Oracle DW	Amazon Red	Améliorations fonctionnelles mises en œuvre pour le traitement approximatif des requêtes, telles que :  Fonctions d'agrégation <ul style="list-style-type: none"><li>• ANY_VALUE</li><li>• APPROX_COUNT_DISTINCT</li></ul>

Source	Cible	Nouveautés, améliorations ou correctifs
		<ul style="list-style-type: none"> <li>• APPROX_COUNT_DISTINCT_DETAIL</li> <li>• APPROX_COUNT_DISTINCT_AGG</li> <li>• LISTAGG</li> <li>• TO_APPROX_COUNT_DISTINCT</li> </ul>
Teradata	Amazon Red	Améliorations de conversion mises en œuvre pour le tri automatique Teradata et la sélection des clés de distribution. Le moteur de base de données sélectionne automatiquement les clés de distribution et de tri. Ajout d'un bouton radio intitulé Utiliser le réglage automatique des tableaux d'Amazon Redshift dans la boîte de dialogue Paramètres des projets en cours > Stratégies d'optimisation > Stratégie de sélection initiale des clés.
Teradata	Amazon Red	Chargeur de AWS SCT tables amélioré pour garantir le AWS SCT chargement de toutes les tables depuis Teradata.
Teradata	Amazon Red	Améliorations de conversion mises en œuvre afin qu'Amazon Redshift prenne en charge les modèles de sous-requêtes corrélés qui incluent une simple clause WHERE NOT EXISTS.
Teradata	Amazon Red	Ajout du support pour l'utilisation des commandes ECHO dans les macros.
DB2 LUW	PostgreSQL	Support mis en œuvre pour la conversion des ENSEMBLES DE RÉSULTATS DYNAMIQUES, notamment :
	Aurora PostgreSQL	<ul style="list-style-type: none"> <li>• Clause de curseur AVEC RETOUR/AVEC RETOUR AU CLIENT</li> <li>• Conversion des clauses de routine DYNAMIC RESULT SETS</li> </ul>

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Aurora PostgreSQL	Support implémenté pour la version actuelle d'Aurora RDS PostgreSQL en tant que cible.
Oracle	L	
DB2 LUW		
SAP ASE		
Microsoft SQL Server	MariaDB	Support implémenté pour MariaDB 10.5 en tant que cible.
Oracle		
DB2 LUW		
SAP ASE		
Microsoft SQL Server	MariaDB	Support implémenté de INSERT-RETURNING qui renvoie un ensemble de résultats des lignes insérées.
Oracle	Aurora PostgreSQL	Ajout du support de la fonction XMLFOREST pour la conversion d'Oracle vers Aurora PostgreSQL.
	L	

#### Problèmes résolus :

- Améliorations générales

# Notes de mise à jour pour la AWS SCT version 648

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL L  Amazon Aurora PostgreSQL L-Compatible Edition	Mode d'application personnalisé du pack d'extension Aurora PostgreSQL implémenté : opérateurs pour les types numérique/date et texte.
Oracle  Microsoft SQL Server  DB2 LUW	Aurora PostgreSQL	Configuration d'Aurora PostgreSQL Lambda Invoke implémentée : création de l'extension aws_lambda ; attribution du rôle IAM au cluster Aurora PostgreSQL. <ul style="list-style-type: none"> <li>• Oracle : e-mails, tâches, files d'attente, fichiers WebAgent</li> <li>• DB2—E-mails, tâches, fichiers</li> <li>• Microsoft SQL Server : e-mails, agent</li> </ul>
Oracle	PostgreSQL L	La refactorisation de la conversion des instructions FORALL a été mise en œuvre : <ul style="list-style-type: none"> <li>• Déclaration FORALL</li> <li>• POUR TOUS... ENREGISTRER LES EXCEPTIONS</li> <li>• RETOUR AVEC BULK COLLECT</li> <li>• Collection du système SQL%BULK_EXCEPTIONS</li> </ul>



Source	Cible	Nouveautés, améliorations ou correctifs
Oracle DW 18, 19	Amazon Red	Améliorations apportées à la conversion d'Oracle vers Amazon Redshift : intégration améliorée de la conversion. LISTAGG agrégé ; LISTAGG analytique.
Oracle DW 18,19	Amazon Red	Améliorations apportées à la conversion d'Oracle vers Amazon Redshift : recherchez de nouvelles fonctionnalités.
Vertica	Amazon Red	Améliorations apportées à la conversion de Vertica vers Amazon Redshift : connexion SSL vers JDBC avec SSL=true.
Microsoft SQL Server DW	Amazon Red	Améliorations apportées à la conversion de Microsoft SQL Server vers Redshift : tables externes.
Teradata	Amazon Red	Améliorations de la conversion de Teradata vers Redshift : opérations arithmétiques sur les types de données INTERVAL.
Teradata	Amazon Red	Améliorations de la conversion de Teradata vers Redshift : Support pour les alias de colonnes latérales.

Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 647

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SQL Server	Microsoft SQL Server	RDS prend désormais en charge la fonctionnalité Database Mail.
Microsoft SQL Server	MySQL	Implémentation du nom maximal pour chaque type d'identifiant : la longueur maximale des noms d'objets (par exemple, tables, contraintes, colonnes) dans SQL Server est de 128 caractères. La longueur

Source	Cible	Nouveautés, améliorations ou correctifs
		<p>maximale des noms d'objets dans MySQL est de 64 caractères. Pour écrire des objets convertis dans la base de données MySQL, vous devez raccourcir leur nom. Pour éviter les doublons après la découpe, vous devez ajouter la « somme de contrôle » du nom de l'objet d'origine aux nouveaux noms.</p> <p>Coupez les noms de plus de 64 caractères comme suit :</p> <pre>[first N chars]() + "" + [checksum]()</pre> <pre>[first N chars] = 64 - 1 - [length of checksum string]</pre> <p>Par exemple :</p> <pre>example_of_a_test_schema_with_a_name_length_g reater_than_64_characters ?? example_of_a_test _schema_with_a_name_length_greater_than_64_97 03</pre>
Oracle	MySQL/ Aurora MySQL	Implémentation du chargement et de la conversion des commentaires sur les objets de stockage. Par exemple, le traitement des commentaires sur les tables et le traitement des commentaires sur les colonnes Table/Vue.
Teradata	Amazon Red	Ajout du support pour la conversion des types de données TIME.
Teradata	Amazon Red	Améliorations de conversion — TD_NORMALIZE_OVERLAP a été implémenté.
Microsoft SQL Server DW	Amazon Red	Améliorations de conversion — SELECT avec clause WITH ; SELECT sans FROM
Tous	Tous	AWS SCT Data Migration Service Assessor (DMSA) : cette nouvelle fonctionnalité vous permet d'évaluer plusieurs serveurs et de recevoir un rapport récapitulatif indiquant la meilleure orientation cible pour votre environnement.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	AWS SCT Assistant — La comparaison des cibles affiche désormais les différences entre les cibles dans une seule vue tabulaire.
Tous	Tous	Interface utilisateur du filtre arborescent : le filtre de métadonnées repensé gère des modèles de filtrage plus complexes.
Tous	Tous	Rapport d'évaluation — La section Avertissement repensée fournit une meilleure description et une meilleure compréhension d'un problème.

### Problèmes résolus :

- Améliorations générales
- Extracteurs de données — La sous-tâche a échoué avec `ConcurrentModificationException` [RÉSOLU].
- Microsoft SQL Server vers MySQL : longueur maximale des identifiants [RÉSOLU].

## Notes de mise à jour pour la AWS SCT version 646

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL	Implémentation améliorée du modèle de format TM.
Oracle	PostgreSQL	L'implémentation du masque au format SP fournit un support de base pour le suffixe SP, uniquement pour la langue anglaise.
Oracle	PostgreSQL	Gestion des noms d'objets longs Oracle : gère AWS SCT désormais les noms d'objets longs Oracle en fonction de l'attribut de longueur maximale de l'identifiant cible.
	Amazon Redshift	Encodage Amazon Redshift AZ64 avec AWS SCT — Ajout d'un codage de compression AZ64 pour certains types de données

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Ajout du support pour la conversion des transactions implicites.
Teradata	Amazon Red	Ajout de la prise en charge des fonctions géospatiales intégrées de Teradata : Méthodes ST_LineString
Greenplum	Amazon Red	Conversion de séquences Greenplum — Ajout des éléments suivants aux onglets Propriétés : valeur minimale, valeur maximale, incrément , cycle.
Greenplum	Amazon Red	Résolveur — Ajout de la résolution du type de données « char ».
Greenplum	Amazon Red	Longueur de conversion des caractères — Conversion PL/pgSQL mise à jour pour le type de caractère.
Greenplum	Amazon Red	Résolution d'un problème lié à la sélection de la clé de distribution par Greenplum, à cause duquel une table possédait une CLÉ DE DISTRIBUTION mais AWS SCT ne parvenait pas à reconnaître et à récupérer la table comme étant DISTRIBUÉE ALÉATOIREMENT.
Teradata	Amazon Red	Support des curseurs Teradata — Ajout de la prise en charge de la conversion des curseurs.
Teradata	Amazon Red	Colonnes d'identité — Ajout du support pour la conversion des colonnes d'identité.
Teradata	Amazon Red	Types de données INTERVAL — Ajout de la prise en charge de la conversion des types de données INTERVAL.

#### Problèmes résolus :

- Améliorations générales
- Greenplum : Impossible d'exécuter la conversion en raison d'une erreur dans le journal [RÉSOLU].
- MSSQL — PostgreSQL : erreur du transformateur lors de la conversion de la fonction LAG [RÉSOLU].
- MSSQL — PostgreSQL : SCOPE\_IDENTITY [RÉSOLU].
- AWS SCT accrochage dans les projets DW [RÉSOLU].

- Besoin d'une règle de mappage pour supprimer de l'espace supplémentaire sur le nom de colonne dans AWS SCT [RESOLVED].

## Notes de mise à jour pour la AWS SCT version 645

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Fournissez une solution pour résoudre les vues non entièrement qualifiées de Teradata (noms de vues ou objets non entièrement qualifiés dans la vue).
Teradata	Amazon Red	Ajout du support de la fonction ASCII pour les nœuds de calcul.
Teradata	Amazon Red	Lorsque AWS SCT des données multioctets sont détectées dans un Teradata CHAR défini comme telCHAR(N), elles sont converties VARCHAR(3*N) dans Amazon Redshift.
Teradata	Amazon Red	Fournissez la CAST conversion Teradata entre les dates et les nombres. <ul style="list-style-type: none"> <li>• <code>SELECT Cast('2020-07-17' AS BIGINT)</code></li> <li>• <code>SELECT Cast(20200630 - 19000000 AS DATE)</code></li> </ul>
Teradata	Amazon Red	Support de conversion des types de PERIOD données Teradata en deux colonnes Amazon TIMESTAMP Redshift : <ul style="list-style-type: none"> <li>• <code>PERIOD(TIMESTAMP)</code></li> <li>• <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code></li> </ul>
Teradata	Amazon Red	Support de conversion de la RANK fonction Teradata avec RESET WHEN clause.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Support amélioré de CAST dans les conversions de types de données explicites et implicite dans CASTs les expressions.
Teradata	Amazon Red	Signalez les modèles de sous-requêtes corrélés non pris en charge. Pour plus d'informations, consultez la section <a href="#">Sous-requêtes corrélées</a> dans le manuel Amazon Redshift Database Developer Guide.
Aucun(e)	Amazon Red	Les tables améliorées limitent la prise en charge des types de RA3 nœuds.
Teradata	Amazon Red	Ajout de la prise en charge de l'extraction de données géospatiales Teradata. Pour plus d'informations, consultez la section <a href="#">Interrogation de données spatiales dans Amazon Redshift dans le manuel Amazon Redshift Database Developer Guide</a> .
Microsoft SQL Server	PostgreSQL	Ajout de l'option, <code>convert_procedures_to_function</code> .

#### Problèmes résolus :

- Améliorations générales

## Notes de mise à jour pour la AWS SCT version 644

Les modifications apportées aux AWS SCT versions 1.0.643 sont fusionnées dans la AWS SCT version 1.0.644.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Améliorations multiples en matière de conversion. <ul style="list-style-type: none"> <li>• Conversions améliorées QUALIFY grâce à l'alias de table.</li> <li>•</li> </ul>

Source	Cible	Nouveautés, améliorations ou correctifs
		<p>Conversions améliorées avec l'INopérateur.</p> <ul style="list-style-type: none"> <li>• Conversion améliorée avec l'LIKEopérateur.</li> <li>• Conversions améliorées avec mise en évidence des problèmes dans le code converti.</li> <li>• Conversions améliorées avec des QUALIFY clauses WHERE d'ordre inhabituel dans SQL.</li> <li>• Des erreurs de transformation corrigées se sont produites lors JOIN( ) de la conversion de la procédure de constructionUPD_FT_SVC_TRANS_BH_CBH_IND .</li> <li>• Conversion améliorée des macros en procédures stockées.</li> </ul> <p>Ajout de commandes AWS SCT CLI spéciales qui peuvent analyser les scripts sql/bteq fournis et générer un rapport sur le nombre de structures syntaxiques rencontrées dans le code source.</p> <ul style="list-style-type: none"> <li>• Nombre de commandes BTEQ</li> <li>• Comte de MANUTENTIONNAIRES</li> <li>• Nombre de cas CAST</li> <li>• Nombre de cas DML/DDDL</li> <li>• Vous pouvez compter DMLs sur des vues actualisables</li> </ul> <p>Ajout d'un élément d'action du rapport d'évaluation : les colonnes Teradata avec des formats de date personnalisés ne sont pas prises en charge dans Amazon Redshift.</p>

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Fonctionnalité ajoutée pour enregistrer les scripts d'installation des packs d'extension.</p> <p>Niveau de gravité modifié pour AI 5334.</p> <p>Amélioration des performances d'utilisation d'un enregistrement comme variable de package <code>IMPLEMENTATION</code> .</p> <p>Ajout du support de la fonction d'XMLAGGagrégation</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	<p>Ajout du chargement et de la conversion des commentaires sur l'implémentation des objets de stockage.</p>
MS SQL DW	Amazon Red	<p>Amélioration de la conversion : problème résolu avec <code>PATINDEX</code>.</p> <p>Améliorations de l'interface :</p> <ul style="list-style-type: none"> <li>• Enregistrez au format SQL pour l'implémentation de l'arborescence des sources.</li> <li>• Ajout d'une logique supplémentaire à la génération de scripts pour plusieurs fichiers.</li> </ul>
Vertica	Amazon Red	<p>Amélioration de l'interface utilisateur : enregistrez au format SQL pour l'implémentation de l'arborescence des sources.</p>

#### Problèmes résolus :

- Améliorations générales apportées aux conversions entre Teradata et Amazon Redshift
- Correction de bogues généraux et améliorations de l'interface utilisateur

## Notes de mise à jour pour la AWS SCT version 642

Changements apportés à la AWS Schema Conversion Tool version 1.0.642.



**Note**

AWS Schema Conversion Tool (AWS SCT) les modifications apportées à la version 1.0.642 sont applicables à Windows, Ubuntu et Fedora. Il n'existe pas de version 1.0.642 pour macOS.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft SSIS	AWS Glue	Mise en œuvre de la conversion des packages ETL Microsoft SQL Server Integration Services (SSIS) en AWS Glue. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de SSIS en AWS Glue with AWS SCT</a> .
Oracle	MariaDB/SQL MODE=C LE/MySQL/ AmazonAurora MySQL	Implémentation de la section de déclaration PL/SQL dans la clause WITH.
Oracle	PostgreSQL/Aurora PostgreSQL	Ajout de la prise en charge d' <code>DBMS_SESSION.RESET_PACKAGE</code> et <code>DBMS_SESSION.MODIFY_PACKAGE</code> .
Vertica	Amazon Redshift	Activez l'exportation de scripts SQL depuis une base de données Vertica vers Amazon Redshift.

## Problèmes résolus :

- Amélioration du rapport d'évaluation.
- Amélioration de l'interface utilisateur du rapport d'évaluation.
- Ajoutez la possibilité de modifier les paramètres JVM depuis l'interface utilisateur.
- Améliorations générales

# Notes de mise à jour pour la AWS SCT version 641

Changements apportés à la AWS Schema Conversion Tool version 1.0.641.

## Note

AWS Schema Conversion Tool (AWS SCT) les modifications apportées à la version 1.0.641 sont applicables à Windows, Ubuntu et Fedora. Il n'existe pas de version 1.0.641 pour macOS.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle/ MS SQL/ MySQL/ PostgreS QL/Db2 LUW	Tous	Produisez les calculs du rapport temporel dans le fichier .csv.
Teradata	Amazon Red	Ajout du support pour la fonction CSUM.  Ajout de la prise en charge des types de données géospatiales Teradata.
Teradata	Tous	Ajout du support pour la conversion des colonnes IDENTITY.
Greenplum	Amazon Red	Ajout du support pour le style de distribution AUTO lors de la conversion des tables Greenplum.
SAP ASE	Tous	Produisez les calculs du rapport temporel dans le fichier .csv.

Résolution :

- Correction de divers bogues
- Diverses améliorations des performances.

# Notes de mise à jour pour la AWS SCT version 640

Les modifications apportées aux AWS SCT versions 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 et 1.0.640 sont fusionnées dans la version 1.0.640. AWS SCT

## Note

AWS SCT Les modifications apportées à la version 1.0.640 sont applicables à Windows, Ubuntu et Fedora. Ils ne s'appliquent pas aux macOS.

Vous ne pouvez pas installer AWS SCT la version 1.0.640 ou supérieure sur Apple macOS. AWS SCT la version 1.0.632 était la dernière version à prendre en charge l'installation sur Apple macOS.

Dans les tableaux suivants, vous trouverez les listes des fonctions et des correctifs de bogues pour les versions AWS Schema Conversion Tool combinées dans la version 1.0.640. Ces tableaux regroupent les fonctions et les correctifs de bogues en fonction du moteur source.

## Rubriques

- [Modifications apportées à Oracle dans la version 1.0.640](#)
- [Modifications apportées à la version 1.0.640 de Microsoft SQL Server](#)
- [Modifications de la version 1.0.640 pour MySQL](#)
- [Modifications apportées à PostgreSQL dans la version 1.0.640](#)
- [Modifications apportées à la version 1.0.640 de DB2 LUW](#)
- [Modifications apportées à Teradata dans la version 1.0.640](#)
- [Modifications apportées à la version 1.0.640 pour les autres moteurs](#)

## Modifications apportées à Oracle dans la version 1.0.640

Le tableau suivant répertorie les modifications du build 1.0.640 dans lesquelles Oracle est le moteur source.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL	
	L	

Source	Cible	Nouveautés, améliorations ou correctifs
	Aurora PostgreSQL L	Conversion de code SQL implémentée dans les applications Java et Pro*C.
Oracle	PostgreSQL L  Aurora PostgreSQL L	Amélioration des performances des fonctions suivantes lorsqu'elles sont utilisées dans une clause WHERE : <ul style="list-style-type: none"> <li>• aws_oracle_ext.to_date</li> <li>• aws_oracle_ext.to_char</li> <li>• aws_oracle_ext.to_number</li> <li>• aws_oracle_ext.sysdate</li> <li>• aws_oracle_ext.sys_context</li> </ul>
Oracle	RDS	Ajout de la prise en charge de RDS MariaDB 10.4 pour tous les fournisseurs de traitement transactionnel en ligne (OLTP).
Oracle	PostgreSQL L/Aurora PostgreSQL L	Ajout de la prise en charge de DBMS_UTILITY.GET_TIME. Ajout des émulations suivantes : <ul style="list-style-type: none"> <li>• DBMS_UTILITY.GET_TIME</li> <li>• DBMS_UTILITY.FORMAT_CALL_STACK</li> <li>• DBMS_UTILITY.CURRENT_INSTANCE</li> </ul>

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	MariaDB/MySQL/Aurora MySQL/Microsoft SQL Server Mode=Oracle/PostgreSQL/Aurora PostgreSQL/RDSOracle	Ajout de la prise en charge des clauses de partage pour TABLE(DATA,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) et SEQUENCE(DATA)
Oracle	PostgreSQL/Aurora PostgreSQL/OracleRDS	<p>La définition DEFAULT d'une colonne peut être étendue pour que la clause DEFAULT soit appliquée pour une insertion NULL explicite.</p> <p>La clause DEFAULT a une nouvelle clause ON NULL. Cette nouvelle clause demande à la base de données d'affecter une valeur de colonne par défaut spécifiée lorsqu'une instruction INSERT tente d'affecter une valeur qui correspond à NULL.</p>
Oracle	MariaDB/MariaDB (MODE SQL=ORACLE)	Ajout de la prise en charge des « colonnes d'identité », qui s'incrémentent automatiquement au moment de l'insertion.
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
Oracle	MariaDB 10.2/ MariaDB 10.3/PostgreSQL MySQL/ Aurora MySQL/ Pos greSQL/Aurora	La clause DEFAULT comporte une nouvelle clause ON NULL, qui indique à la base de données d'affecter une valeur de colonne par défaut spécifiée lorsqu'une instruction INSERT tente d'affecter une valeur qui correspond à NULL.
Oracle	Oracle RDS/ MySQL /Aurora MySQL/ Pos greSQL/ Aurora PostgreSQL	Ajout de la prise en charge des colonnes IDENTITY.
Oracle	MySQL 8.x	Ajout de la prise en charge de la contrainte CHECK.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Mise en œuvre de la vérification que ANYDATA EST NULL/IS NOT NULL à l'aide de la routine du pack d'extension.</p> <p>Implémentation de l'émulation de la fonction VALUE utilisée dans une requête basée sur la fonction TABLE de XMLSequence.</p> <p>Ajout de la prise en charge de DBMS_LOB pour les routines intégrées suivantes :</p> <ul style="list-style-type: none"> <li>• DBMS_LOB.CREATETEMPORARY</li> <li>• DBMS_LOB.FREETEMPORARY</li> <li>• DBMS_LOB.APPEND</li> </ul>
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.</p>
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	Amazon Red	<p>Mise en œuvre de la conversion des attributs de curseur dans les blocs imbriqués.</p> <p>Amazon Redshift ne prend pas en charge les collections. Les variables associées sont converties en VARCHAR. Toutes les opérations de collecte autres que l'attribution d'une variable à une autre sont rejetées, y compris l'accès aux éléments de collecte et d'initiation.</p> <p>Style de distribution Amazon Redshift implémenté = AUTO.</p>



Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL L/Aurora PostgreSQL L	<p>Si un mot non réservé dans Oracle est réservé dans PostgreSQL, ce qui suit est vrai :</p> <ul style="list-style-type: none"> <li>• Si le mot est entre guillemets, il conserve sa casse et reste cité.</li> <li>• Si le mot n'est pas entre guillemet, il est converti en majuscules et entouré de guillemets.</li> </ul> <p>Mise en œuvre de la possibilité d'utiliser des fonctions comme entrée pour les fonctions LTRIM, RTRIM et TRIM.</p> <p>Les expressions SELECT DISTINCT, ORDER BY doivent apparaître dans la liste de sélection.</p> <p>Pour les paramètres de curseur qui suivent un paramètre avec une valeur DEFAULT, AWS SCT ajoute la clause DEFAULT IS NULL</p> <p>Les paramètres du curseur source OUT sont convertis en paramètres de curseur IN.</p> <p>Nouvelle mise en œuvre de la variable de package en ajoutant l'option « Mise en œuvre logique des variables de package » sous « Paramètres de conversion ». Les paramètres disponibles sont : « variables de session » et « objets globaux plv8 ». La valeur par défaut est « variables de session ».</p> <p>Mise en œuvre de la prise en charge du pragma AUTONOMOUS_TRANSACTION avec dblink et pg_background.</p>
Oracle	Tous	Mise en œuvre de la vue SYS_%_TAB_COMMENTS.

Source	Cible	Nouveautés, améliorations ou correctifs
Oracle	PostgreSQL	Les entrées variables dans les filtres ne sont pas prises en charge dans PostgreSQL. Lors de la conversion depuis Oracle vers PostgreSQL, si un filtre de variable est détecté, une exception est désormais signalée.
Oracle	Amazon Redshift	<p>Mise en œuvre du code stocké FOR..LOOP Amélioration de la conversion du curseur.</p> <p>Mise en œuvre de l'invocation de code stocké de fonction/procédures avec les paramètres par défaut.</p> <p>Mise en œuvre de la capacité du code stocké à effectuer une opération de mise à jour (UPDATE) avec alias sans clause WHERE.</p> <p>Les fonctions de code stocké implémentées préforment les casses supplémentaires avec SELECT FROM dual.</p> <p>Mise en œuvre des paramètres et des variables de package %ROWTYPE de code stocké.</p> <p>Mise en œuvre du code stocké utilisé pour JAVA et les procédures externes.</p> <p>Mise en œuvre du package Oracle standard dans le code stocké.</p>

## Modifications apportées à la version 1.0.640 de Microsoft SQL Server

Le tableau suivant répertorie les modifications du build 1.0.640 dans lesquelles Microsoft SQL Server est le moteur source.

Source	Cible	Nouveautés, améliorations ou correctifs
Microsoft Azure/ Microsoft	PostgreSQL/Aurora PostgreSQL/MySQL/	Ajout de la prise en charge des index COLUMN STORE.

Source	Cible	Nouveautés, améliorations ou correctifs
SQL Server	Aurora MySQL	
Microsoft SQL Server	RDS MariaDB 10.4	Ajout de la prise en charge de RDS MariaDB 10.4 pour tous les fournisseurs de traitement transactionnel en ligne (OLTP).
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Ajout de la prise en charge de l'attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Ajout de la prise en charge des types de table de base de données et Edge.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Ajout de la prise en charge de TEMPORAL TABLES.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQL/ MariaDB	Ajout de la prise en charge du traitement DML pour l'architecture graphique de SQL Server.
SQL Server	Aurora PostgreSQL	Ajout d'une option pour convertir les paramètres sans le préfixe <code>par_</code> .
Azure/SQL Server	MySQL 8.x	Ajout de la prise en charge de la contrainte CHECK.
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index <code>OPTIMIZE_FOR_SEQUENTIAL_KEY</code>.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de <code>TEMPORAL TABLES</code>.</p>

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.
SQL Server	AWS Glue (coque en Python)	Améliorations de conversion, notamment : <ul style="list-style-type: none"> <li>• Mise en œuvre de la conversion des fonctions intégrées en Python.String.</li> <li>• Mise en œuvre EXECUTE et EXEC dans le code stocké.</li> <li>• Mise en œuvre avec les types de table.</li> </ul>
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Mise en œuvre rendant les procédures \$TMP facultatives.
SQL Server	MySQL/Aurora MySQL	Opérations arithmétiques étendues avec dates. Émulation de construction pour TOP (expression) WITH TIES. Après l'appel des procédures avec la variable refcursor générée out, le refcursor se ferme maintenant. La définition d'un niveau d'isolement GLOBAL n'est pas prise en charge dans Aurora MySQL. Seule la portée de la session peut être modifiée. Le comportement par défaut des transactions consiste à utiliser les lectures répétées (REPEATABLE READ) et les lectures constantes. Il peut être nécessaire de modifier les applications conçues pour être utilisées avec READ COMMITTED. Alternativement, elles peuvent remplacer explicitement la valeur par défaut par READ COMMITTED.

Source	Cible	Nouveautés, améliorations ou correctifs
SQL Server	AWS Glue (coque en Python)	<p>Les instructions SQL Server produisent un jeu de résultats complet, mais certains scénarios privilégient un traitement des résultats une ligne à la fois. L'ouverture d'un curseur au niveau d'un jeu de résultats permet de traiter ce jeu de résultats une ligne à la fois. Vous pouvez affecter un curseur à une variable ou à un paramètre avec un type de données de curseur.</p> <p>Implémenté en incluant une série d'instructions Transact-SQL pour le code stocké afin qu'un groupe d'instructions Transact-SQL puisse être exécuté même si Python ne prend pas en charge les fonctions BEGIN et END as de SQL Server. control-of-flow</p> <p>Les instructions SQL Server LABEL et GOTO ne sont pas prises en charge par AWS Glue. Si AWS SCT identifie une instruction LABEL dans le code, elle est ignorée. Si AWS SCT identifie une instruction GOTO, elle est commentée.</p>
SQL Server	Amazon Red	<p>Mise en œuvre du traitement conditionnel des instructions Transact-SQL pour le code stocké en implémentant le contrôle IF ... ELSE.</p> <p>Implémenté en incluant une série d'instructions Transact-SQL pour le code stocké afin qu'un groupe d'instructions Transact-SQL puisse être exécuté sous forme de bloc. Prend en charge les blocs BEGIN ... END imbriqués.</p> <p>Mise en œuvre de SET et SELECT dans le code stocké.</p> <p>A implémenté CREATE INDEX dans Amazon Redshift (qui ne prend pas en charge les index) en créant une clé de tri spécifiée par l'utilisateur sur les tables.</p>

## Modifications de la version 1.0.640 pour MySQL

Le tableau suivant répertorie les modifications du build 1.0.640 dans lesquelles MySQL est le moteur source.

Source	Cible	Nouveautés, améliorations ou correctifs
MySQL	PostgreSQL L 12.x	Ajout de la prise en charge des colonnes générées.
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
MySQL	PostgreSQL L/Aurora PostgreSQL L 11.	Ajout de la prise en charge des éléments suivants : <ul style="list-style-type: none"> <li>• Transactions incorporées dans les procédures stockées SQL.</li> <li>• Possibilité d'appeler des procédures stockées SQL.</li> <li>• Possibilité de créer des procédures stockées SQL.</li> </ul>
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.</p>
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.

## Modifications apportées à PostgreSQL dans la version 1.0.640

Le tableau suivant répertorie les modifications du build 1.0.640 dans lesquelles PostgreSQL est le moteur source.

Source	Cible	Nouveautés, améliorations ou correctifs
PostgreSQL	MySQL 8.x	<p>MySQL prend désormais en charge la création de parties de clé d'index fonctionnelles qui indexent les valeurs d'expression plutôt que les valeurs de colonne. Les parties de clés fonctionnelles permettent l'indexation de valeurs (JSON, par exemple), qui ne peuvent normalement pas être indexées.</p> <p>MySQL prend désormais en charge Now CTE et Recursive CTE.</p>
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>Ajout de la prise en charge des éléments suivants :</p> <ul style="list-style-type: none"> <li>• Transactions incorporées dans les procédures stockées SQL.</li> <li>• Possibilité d'appeler des procédures stockées SQL.</li> <li>• Possibilité de créer des procédures stockées SQL.</li> </ul>
PostgreSQL	MySQL 8.x	<p>Ajout de la prise en charge de MySQL pour les index par ordre décroissant. DESC dans une définition d'index n'est plus ignoré, mais provoque le stockage des valeurs clés dans l'ordre décroissant.</p> <p>Ajout de la prise en charge de MySQL pour l'utilisation d'expressions comme valeurs par défaut dans les spécifications de type de données, y compris les expressions comme valeurs par défaut pour les types de données BLOB, TEXT, GEOMETRY et JSON.</p> <p>Plusieurs fonctions d'agrégation existantes peuvent désormais être utilisées en tant que fonctions de fenêtrage :</p> <ul style="list-style-type: none"> <li>• AVG()</li> </ul>



Source	Cible	Nouveautés, améliorations ou correctifs
		<ul style="list-style-type: none"><li>• BIT_AND()</li><li>• BIT_OR()</li><li>• BIT_XOR()</li><li>• COUNT()</li><li>• JSON_ARRAYAGG()</li><li>• JSON_OBJECTAGG()</li><li>• MAX()</li><li>• MIN()</li><li>• STDDEV_POP()</li><li>• STDDEV()</li><li>• STD()</li><li>• STDDEV_SAMP()</li><li>• SUM()</li><li>• VAR_POP()</li><li>• VARIANCE()</li><li>• VAR_SAMP()</li></ul> <p>MySQL prend en charge les fonctions de fenêtrage qui, pour chaque ligne d'une requête, effectuent un calcul à l'aide de lignes liés à cette ligne.</p> <ul style="list-style-type: none"><li>• CUME_DIST()</li><li>• DENSE_RANK()</li><li>• FIRST_VALUE()</li><li>• LAG()</li><li>• LAST_VALUE()</li><li>• LEAD()</li><li>• NTH_VALUE()</li><li>• NTILE()</li></ul>

Source	Cible	Nouveautés, améliorations ou correctifs
		<ul style="list-style-type: none"> <li>• PERCENT_RANK( )</li> <li>• RANK( )</li> <li>• ROW_NUMBER( )</li> </ul>
PostgreSQL	MySQL 8.x	Ajout de la prise en charge de la contrainte CHECK.
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.</p>
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.
PostgreSQL/Aurora PostgreSQL	Tous	<p>Ajout de l'émulation sysindexes des vues système.</p> <p>S'il existe une instruction SELECT dans une procédure sans spécifier INTO, le paramètre INOUT p_refcur de type refcursor est créé pour une procédure sur la cible.</p>

## Modifications apportées à la version 1.0.640 de DB2 LUW

Le tableau suivant répertorie les modifications apportées à la version 1.0.640 dans laquelle DB2 LUW est le moteur source.

Source	Cible	Nouveautés, améliorations ou correctifs
DB2 LUW	RDS	Ajout de la prise en charge de RDS MariaDB 10.4 pour tous les fournisseurs de traitement transactionnel en ligne (OLTP).
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
DB2 LUW	MySQL 8.0.1	Ajout de la prise en charge des contraintes CHECK.
Tous	SQL Server	SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.  SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.  SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.

## Modifications apportées à Teradata dans la version 1.0.640

Le tableau suivant répertorie les modifications apportées à la version 1.0.640 par les moteurs source Teradata.

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Ajout de la prise en charge des instructions MERGE et QUALIFY.

Source	Cible	Nouveautés, améliorations ou correctifs
		<p>Suppression de la clause LOCKING ROWS FOR ACCESS dans les instructions Teradata.</p> <p>Ajout de la prise en charge de la fonction CAST.</p>
Tous	Tous	<p>Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.</p>
Teradata	Teradata	<p>Améliorations mises en œuvre dans REGEXP_INSTR() et REGEXP_SUBSTR().</p>
Tous	Tous	<p>Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.</p>
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.</p>
Teradata	Tous	<p>Ajout de la prise en charge de REGEXP_INSTR() et REGEXP_SUBSTR().</p>
Tous	Tous	<p>Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.</p>

Source	Cible	Nouveautés, améliorations ou correctifs
Teradata	Amazon Red	Mise en œuvre de la possibilité d'enregistrer le code SQL de l'arborescence source dans un seul fichier ou plusieurs fichiers par étape dans les paramètres du projet : Enregistrer sous SQL, Appliquer, puis Liste déroulante : Fichier unique/Plusieurs fichiers.  Améliorations apportées aux conversions de vues et de procédures.
Teradata	Tous	Ajout de la prise en charge de Teradata version 16.20

## Modifications apportées à la version 1.0.640 pour les autres moteurs

Le tableau suivant répertorie les modifications du build 1.0.640 pour les autres moteurs source.

Source	Cible	Nouveautés, améliorations ou correctifs
Sybase	RDS	Ajout de la prise en charge de RDS MariaDB 10.4 pour tous les fournisseurs de traitement transactionnel en ligne (OLTP).
SAP ASE	MariaDB	Mise en œuvre suivantes : <ul style="list-style-type: none"> <li>• MariaDB 10.4</li> <li>• Instruction EXECUTE IMMEDIATE</li> <li>• Définitions DEFAULT</li> <li>• Prise en charge des contraintes CHECK</li> </ul>
SAP ASE	PostgreSQL 12.x	Ajout de la prise en charge des colonnes générées.
Tous	Tous	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Pour plus d'informations, y compris les liens de téléchargement, consultez <a href="#">Qu'est-ce qu'Amazon Corretto 11 ?</a> dans le guide de l'utilisateur d'Amazon Corretto 11.

Source	Cible	Nouveautés, améliorations ou correctifs
Tous	Tous	Ajout d'informations au rapport d'évaluation sur les incohérences possibles dans la base de données de l'utilisateur.
SAP ASE	MySQL 8.0.1	Ajout de la prise en charge des contraintes CHECK.
Tous	SQL Server	<p>SQL Server 2019 : ajout de la prise en charge du nouvel attribut d'index OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017 : ajout de la prise en charge des types de table de base de données Graph Node et Edge.</p> <p>SQL Server 2016 : ajout de la prise en charge de TEMPORAL TABLES.</p>
Vertica	Amazon Red	Ajout de la prise en charge du style de distribution = AUTO.
Tous	Tous	Implémentation de la possibilité de remplacer les partitions physiques par des partitions virtuelles. Les extracteurs d'entrepôt de données extraient les données en fonction des partitions virtuelles créées.
Amazon Red	Amazon Red	Les fonctions intégrées non prises en charge dans les instructions DML sont remplacées par NULL en tant qu'espace réservé.
Sybase	PostgreSQL	Ajout de la prise en charge des fonctions natives.
SAP ASE	MySQL/ Aurora MySQL	Le niveau d'isolement par défaut pour Aurora MySQL est REPEATABLE READ. La définition d'un niveau d'isolement GLOBAL n'est pas prise en charge dans Aurora MySQL. Seule la portée de la session peut être modifiée. Le comportement par défaut des transactions consiste à utiliser les lectures répétées (REPEATABLE READ) et les lectures constantes. Il peut être nécessaire de modifier les applications conçues pour fonctionner avec READ COMMITTED. Vous pouvez également remplacer explicitement la valeur par défaut par READ COMMITTED.

Source	Cible	Nouveautés, améliorations ou correctifs
SAP ASE	PostgreSQL	Ajout du support pour la fonction CONVERT (optimiste) sans le pack d'extension.
SAP ASE	Tous	Ajout de l'émulation sysindexes des vues système.  S'il existe une instruction SELECT dans une procédure sans spécifier INTO, le paramètre INOUT p_refcur de type refcursor est créé pour une procédure sur la cible.
Greenplum	Amazon Red	Mise en œuvre de CREATE TEMPORARY TABLE comme suit :

# Nouvelles fonctionnalités du AWS Schema Conversion Tool

Le tableau suivant décrit les modifications importantes apportées au guide de l'utilisateur AWS Schema Conversion Tool (AWS SCT) après janvier 2018.

Vous pouvez vous abonner à un flux RSS afin de recevoir les notifications des mises à jour de cette documentation.

Modification	Description	Date
<a href="#">AWS SCT version #1 .0.672</a>	La version 1.0.672 prend en charge Amazon RDS pour PostgreSQL 15 en tant que cible et Microsoft SQL Server version 2022 en tant que source. Il ajoute également la prise en charge des nouvelles fonctionnalités d'Amazon Redshift dans le code converti, met en œuvre plusieurs améliorations de conversion pour la source IBM Db2 for z/OS et résout un certain nombre de problèmes de conversion.	8 mai 2023
<a href="#">AWS SCT version #1 .0.671</a>	La version 1.0.671 prend en charge les migrations d'Apache Oozie vers AWS Step Functions. Il ajoute également un support BigQuery en tant que source pour le processus d'évaluation multiserveur. En outre, il ajoute de nouveaux paramètres de conversion pour IBM Db2 for z/OS en tant que source et	8 mars 2023



résout un certain nombre de problèmes de conversion.

### [AWS SCT version #1 .0.670](#)

La version 1.0.670 prend en charge les migrations de Hadoop vers Amazon EMR. Il ajoute également la prise en charge d'Azure Synapse Analytics en tant que source pour le processus d'évaluation multiserveur. En outre, il améliore la conversion du code SQL intégré aux applications Java et résout un certain nombre de problèmes de conversion.

23 janvier 2023

### [AWS SCT version #1 .0.669](#)

La version 1.0.669 implémente le support du partitionnement natif pour la migration des données depuis les entrepôts de données Oracle. Il améliore également le processus d'évaluation multiserveur, ajoute de nouvelles fonctionnalités aux agents d'extraction de données et résout un certain nombre de problèmes de conversion.

19 décembre 2022

[AWS SCT version #1 .0.668](#)

La version 1.0.668 implémente le partitionnement virtuel automatique pour la migration des données à partir des bases de données Greenplum et ajoute la prise en charge de la migration des données des bases de données Snowflake vers Amazon Redshift. Il améliore également la conversion du code SQL intégré dans les applications C# et résout un certain nombre de problèmes de conversion.

16 novembre 2022

[AWS SCT version #1 .0.667](#)

La version 1.0.667 prend en charge le moteur d'extraction, de transformation et de chargement (ETL) Informatica en tant que source de migration. Il met également à jour la version du pack d'extension, augmente la version minimale du pilote prise en charge pour Amazon Redshift et résout un certain nombre de problèmes de conversion.

13 octobre 2022

[AWS SCT version #1 .0.666](#)

La version 1.0.666 améliore la conversion des applications Java en ajoutant le support du MyBatis framework. Il ajoute également de nouvelles fonctions dans les packs d'extension, améliore le chargeur de métadonnées source et résout un certain nombre de problèmes de conversion.

20 septembre 2022

[AWS SCT version #1 .0.665](#)

La version 1.0.665 fournit le support en BigQuery tant que source de migration . Il implémente également le support de la nouvelle version du fichier de configuration des fonctionnalités de Babelfish. En outre, il améliore la conversion des entrepôts de données vers Amazon Redshift et résout un certain nombre de problèmes de conversion.

29 août 2022

[AWS SCT version #1 .0.664](#)

La version 1.0.664 prend en charge Amazon Redshift Serverless en tant que source ou cible de migration . Il implémente également l'équilibrage automatique de la mémoire dans les tâches d'extraction de données et corrige une erreur en cas d' AWS SCT impossibilité de se connecter aux AWS Snowball Edge appareils. En outre, il permet de modifier le classement des colonnes dans les règles de migration, améliore l'interface utilisateur et résout un certain nombre de problèmes de conversion.

14 juillet 2022

[AWS SCT version #1 .0.663](#)

La version 1.0.663 ajoute le support de Babelfish pour Aurora PostgreSQL 1.2.0 et améliore les capacités des rapports d'évaluation multiserveurs. Il ajoute également de nouvelles fonctionnalités aux règles de migration, corrige deux erreurs de chargement et résout un certain nombre de problèmes de conversion.

20 juin 2022

[AWS SCT version #1 .0.662](#)

La version 1.0.662 implémente la conversion du code SQL dans les applications C# et améliore le flux de travail des rapports d'évaluation multiserveurs. Il ajoute également de nombreuses améliorations de conversion et résout un certain nombre de problèmes de conversion.

19 mai 2022

[AWS SCT version #1 .0.661](#)

La version 1.0.661 prend en charge IBM Db2 for z/OS en tant que source de migration. Il prend également en charge la conversion des scripts d'extraction, de transformation et de chargement (ETL) en scripts AWS Glue Studio et résout un certain nombre de problèmes de conversion.

21 avril 2022

[AWS SCT version #1 .0.660](#)

La version 1.0.660 prend en charge la version majeure 14 de PostgreSQL et MariaDB 10.6 en tant que cibles de migration. Il prend également en charge la conversion des index Oracle en vues matérialisées et résout un certain nombre de problèmes de conversion.

21 mars 2022

[AWS SCT version #1 .0.659](#)

La version 1.0.659 prend en charge la version majeure de PostgreSQL 13 sur Aurora PostgreSQL Compatible Edition en tant que cible de migration. Il implémente la conversion de code SQL dans les applications C#, ajoute le support d'Oracle Unified Auditing et résout un certain nombre de problèmes de conversion.

21 février 2022

[AWS SCT version #1 .0.658](#)

La version 1.0.658 assure l'intégration AWS Secrets Manager et la prise en charge de la plate-forme de base de données cible virtuelle Amazon Redshift. Il ajoute également un certain nombre d'améliorations de conversion et de corrections de bogues.

20 janvier 2022

[AWS SCT version #1 .0.657](#)

La version 1.0.657 améliore la conversion de Microsoft SQL Server vers Aurora PostgreSQL Compatible Edition, Amazon RDS for PostgreSQL et d'autres destinations de migration. Il ajoute également un certain nombre d'améliorations à l'interface utilisateur et de corrections de bogues.

20 décembre 2021

[AWS SCT version #1 .0.656](#)

La version 1.0.656 prend en charge plusieurs bases de données sources et cibles dans un seul projet. Il ajoute également une conversion, une stratégie d'optimisation, des améliorations générales et un certain nombre de corrections de bogues.

22 novembre 2021

[AWS SCT version #1 .0.655](#)

La version 1.0.655 implémente la conversion des scripts de FastExport travail Teradata en Amazon Redshift RSQL et augmente la version de pilote minimale prise en charge pour Greenplum à 42.2.5. Il ajoute également un certain nombre d'améliorations et de corrections de bogues.

18 octobre 2021

[AWS SCT version #1 .0.654](#)

La version 1.0.654 implémente la conversion des scripts Shell, Teradata et Teradata FastLoad Basic Teradata Query (BTEQ) en Amazon Redshift RSQL. Il résout également un certain nombre de problèmes de conversion et ajoute un certain nombre d'améliorations et de corrections de bogues.

16 septembre 2021

[AWS SCT version #1 .0.653](#)

La version 1.0.653 implémente la conversion du SQL dynamique créé dans des fonctions ou des procédures appelées. Il améliore également la conversion des routines cryptées et ajoute un certain nombre d'améliorations et de corrections de bogues.

10 août 2021

[AWS SCT version #1 .0.652](#)

La version 1.0.652 implémente le mode de commande script dans l'interface de ligne de commande et implémente les règles d'optimisation du schéma. Il ajoute également un certain nombre d'améliorations de conversion et de performances ainsi que des corrections de bogues.

30 Juin 2021

[AWS SCT version #1 .0.651](#)

La version 1.0.651 ajoute un certain nombre d'améliorations et de corrections de bugs. Il permet également d'accéder à la copie initiale de la référence AWS Schema Conversion Tool CLI.

4 juin 2021



[AWS SCT version #1 .0.650](#)

La version 1.0.650 implémente le support d'Amazon RDS pour PostgreSQL 13 en tant que base de données cible et met à jour les agents d'extraction. Il met également à niveau la version minimale du pilote JDBC prise en charge pour Microsoft SQL Server, Azure et Azure Synapse. En outre, il ajoute un certain nombre d'améliorations de conversion et de corrections de bogues.

30 avril 2021

[AWS SCT version #1 .0.649](#)

La version 1.0.649 implémente le support de MariaDB 10.5 en tant que base de données cible et implémente des améliorations fonctionnelles pour la conversion des fonctions intégrées d'Oracle. Il ajoute également un certain nombre d'améliorations de conversion et de performances ainsi que des corrections de bogues.

29 mars 2021

[AWS SCT version #1 .0.648](#)

La version 1.0.648 ajoute un certain nombre d'améliorations de conversion et de corrections de bogues.

22 février 2021

[AWS SCT version #1 .0.647](#)

La version 1.0.647 ajoute le support de la fonctionnalité Database Mail sur Amazon RDS, implémente le chargement et la conversion des commentaires sur les objets de stockage. Il ajoute également un évaluateur et un AWS SCT assistant du service de migration des AWS SCT données et implémente l'interface utilisateur du filtre arborescent. En outre, il ajoute une section repensée dans le rapport d'évaluation et un certain nombre d'améliorations et de corrections de bogues.

15 janvier 2021

[AWS SCT version #1 .0.646](#)

La version 1.0.646 ajoute la prise en charge des types de données INTERVAL, des colonnes d'identité et de la conversion des curseurs, et ajoute un certain nombre d'améliorations et de corrections de bogues.

28 décembre 2020

[AWS SCT version #1 .0.645](#)

La version 1.0.645 ajoute le support de l'ETL SSIS pour la AWS Glue conversion et un certain nombre d'améliorations et de corrections de bogues.

16 novembre 2020

<a href="#">AWS SCT build #1 .0.643-1.0.644</a>	La version 1.0.644 ajoute un certain nombre d'améliorations de conversion, de performances et d'interface utilisateur ainsi que des corrections de bogues.	14 octobre 2020
<a href="#">AWS SCT version #1 .0.642</a>	La version 1.0.642 implémente la conversion des packages ETL de Microsoft SQL Server Integration Services vers Microsoft SQL Server Integration Services AWS Glue et ajoute un certain nombre d'améliorations et de corrections de bogues.	28 août 2020
<a href="#">AWS SCT version #1 .0.641</a>	Ajout du support SSL pour les extracteurs de données. La version inclut également un certain nombre d'améliorations et de corrections.	17 juillet 2020
<a href="#">AWS SCT versions #1 .0.633-1.0.640</a>	Mise à niveau de JDK 8 vers Amazon Corretto JDK 11. Ajout de tables identifiant d'autres mises à niveau, modifications et correctifs.	22 juin 2020
<a href="#">Disponibilité d'AWS WQF</a>	AWS SCT ne fournit plus l'outil AWS Workload Qualification Framework (AWS WQF) en téléchargement.	19 juin 2020

[AWS SCT versions #1 .0.632](#)

Interface utilisateur SCT  
- Ajout d'un nouvel onglet pour afficher les erreurs qui se produisent lors de l'application de scripts. Vous pouvez maintenant enregistrer l'arborescence source en tant que SQL lors de la conversion depuis SAP ASE. Améliorations pour les conversions en PostgreSQL ou Aurora PostgreSQL ou Redshift.

19 novembre 2019

[AWS SCT versions #1 .0.631 et #1 .0.630 \(combinées\)](#)

Meilleure prise ROWIDs en charge dans Oracle et pour les objets système dans Microsoft SQL Server et SAP ASE. Meilleure gestion des spécificateurs manquants de schémas SQL Server. Meilleure prise en charge des conversions de Greenplum vers Redshift. Support amélioré pour la conversion du code stocké lors du passage à Amazon Redshift, MariaDB, MySQL et PostgreSQL.

30 septembre 2019

---

<a href="#"><u>AWS SCT version #1 .0.629</u></a>	Prise en charge des procédures stockées pour les conversions depuis Netezza. Support amélioré pour les conversions vers Amazon Redshift, DynamoDB, MySQL et PostgreSQL. Ajout de la prise en charge de SAP ASE 12.5 en tant que source	20 août 2019
<a href="#"><u>AWS SCT version #1 .0.628</u></a>	Support pour l'émulation de service pour les conversions depuis DB2 SQL Server et Oracle. Améliorations apportées aux conversions vers Amazon Redshift, notamment une prise en charge accrue des curseurs et des procédures stockées.	22 juin 2019
<a href="#"><u>AWS SCT version #1 .0.627</u></a>	Support pour les conversions de SQL Server vers des procédures stockées dans Amazon Redshift. Améliorations apportées à la conversion vers PostgreSQL 11 et MySQL 8.0.	31 mai 2019
<a href="#"><u>AWS SCT version #1 .0.626</u></a>	PostgreSQL 11 et MySQL 8.0 sont désormais pris en charge en tant que cibles. SAP ASE 15.5 est maintenant pris en charge en tant que source.	26 avril 2019

[AWS SCT version #1 .0.625](#)

Les mises à jour incluent la possibilité de convertir Teradata BTEQ en AWS Glue, la prise en charge des conversions vers MariaDB 10.3 avec prise en charge du mode de compatibilité Oracle, la prise en charge de SAP ASE 15.7 et les substitutions de services pour émuler les fonctionnalités manquantes.

25 mars 2019

[AWS SCT version #1 .0.624](#)

Les mises à jour incluent la possibilité de convertir Oracle ETL en Amazon AWS Glue RDS for MariaDB et la prise en charge des conversions depuis Microsoft SQL Server, Oracle et IBM Db2 LUW vers Amazon RDS for MariaDB. Nous avons également ajouté la prise en charge des conversions de SAP ASE vers RDS pour MySQL et Amazon Aurora avec la compatibilité MySQL. De plus, nous avons ajouté la prise en charge pour l'extension Oracle au cours de la conversion d'Oracle vers PostgreSQL.

22 février 2019

[AWS SCT version #1 .0.623](#)

Les mises à jour incluent la possibilité de convertir les bases de données SAP ASE et la capacité de conversion des scripts T-SQL, DML et DDL en un code ou des composants équivalents. Nous avons également ajouté les émulations Oracle et Microsoft SQL Server pour améliorer les conversions.

25 janvier 2019

[AWS SCT version #1 .0.622](#)

Les mises à jour incluent l'application Workload Qualification Framework, qui analyse la charge de travail pour une migration complète, y compris les modifications de bases de données et d'application.

20 décembre 2018

[AWS SCT version #1 .0.621](#)

Les mises à jour incluent la prise en charge d'Aurora PostgreSQL 10 en tant que cible, et la possibilité de migrer à partir de Netezza à l'aide d'options de table externe.

21 novembre 2018

[AWS SCT version #1 .0.620](#)

Les mises à jour incluent la possibilité d'enregistrer des scripts SQL et la prise en charge des curseurs globaux lors de la migration vers MySQL.

22 octobre 2018

---

<a href="#">AWS SCT version #1 .0.619</a>	Les mises à jour incluent la prise en charge de la migration depuis Apache Cassandra vers DynamoDB, et la prise en charge de Vertica 9 en tant que source.	20 septembre 2018
<a href="#">AWS SCT version #1 .0.618</a>	Les mises à jour incluent des rapports d'évaluation étendus, la prise en charge de la conversion d'Oracle ROWIDs et la prise en charge des tables définies par l'utilisateur de SQL Server.	24 août 2018
<a href="#">AWS SCT version #1 .0.617</a>	Les mises à jour incluent des rapports d'évaluation étendus, la prise en charge de la conversion d'Oracle ROWIDs et la prise en charge des tables définies par l'utilisateur de SQL Server.	24 juillet 2018
<a href="#">AWS SCT version #1 .0.616</a>	Les mises à jour incluent la prise en charge de RDS lors de la conversion d'Oracle vers Amazon RDS pour Oracle, la conversion d'objets de planification Oracle et la prise en charge des tâches Oracle, du partitionnement de Db2 LUW version 10.1.	26 juin 2018



---

<a href="#">AWS SCT version #1 .0.615</a>	Les mises à jour incluent la prise en charge des instructions GOTO SQL Server vers PostgreSQL, le partitionnement PostgreSQL 10 et Db2 LUW version 10.1.	24 mai 2018
<a href="#">AWS SCT version #1 .0.614</a>	Les mises à jour incluent la prise en charge d'Oracle vers les liens de base de données Oracle, des fonctions en ligne SQL Server vers PostgreSQL et de l'émulation des objets système Oracle.	25 avril 2018
<a href="#">AWS SCT version #1 .0.613</a>	Les mises à jour incluent la prise en charge de Db2 LUW, de la conversion des fichiers SQL*Plus et de l'authentification Windows Microsoft SQL Server.	28 mars 2018
<a href="#">AWS SCT version #1 .0.612</a>	Les mises à jour incluent la prise en charge du mappage de type de données personnalisé, de la comparaison de schémas pour Oracle 10 et de la conversion de variables globales Oracle vers PostgreSQL.	le 22 février 2018

[AWS SCT version #1 .0.611](#)

Les mises à jour incluent la prise en charge de la conversion des instructions dynamiques d'Oracle vers PostgreSQL, de l'ouverture du fichier journal en sélectionnant un message d'erreur et de la possibilité de masquer les schémas dans une arborescence.

23 janvier 2018

## Mises à jour antérieures

Le tableau suivant décrit les modifications importantes apportées au guide de l'utilisateur AWS Schema Conversion Tool (AWS SCT) avant janvier 2018.

Version	Modification	Description	Date de modification
1.0.608	Prise en charge des points de terminaison FIPS pour Amazon S3	Vous pouvez désormais demander AWS SCT à vous connecter à Amazon S3 et Amazon Redshift en utilisant des points de terminaison FIPS afin de vous conformer aux exigences de sécurité de la Federal Information Processing Standard. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des AWS identifiants</a> .	17 novembre 2017
1.0.607	Prise en charge des points de terminaison FIPS pour Amazon S3	Vous pouvez désormais demander AWS SCT à vous connecter à Amazon S3 et Amazon Redshift en utilisant des points de terminaison FIPS afin de vous conformer aux exigences de sécurité de la Federal Information Processing Standard. Pour de plus amples informations, veuillez consulter <a href="#">Stockage des AWS identifiants</a> .	30 octobre 2017

Version	Modification	Description	Date de modification
1.0.607	Les tâches d'extraction de données peuvent être ignorées LOBs	Lorsque vous créez des tâches d'extraction de données, vous pouvez désormais choisir d'ignorer les objets volumineux (LOBs) afin de réduire la quantité de données que vous extrayez. Pour de plus amples informations, veuillez consulter <a href="#">Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données</a> .	30 octobre 2017
1.0.605	Accès au journal des tâches des agents d'extraction de données	Vous pouvez désormais accéder au journal des tâches de l'agent d'extraction de données à partir d'un lien pratique dans l'interface AWS Schema Conversion Tool utilisateur. Pour de plus amples informations, veuillez consulter <a href="#">Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données</a> .	28 août 2017
1.0.604	Améliorations du convertisseur	Le AWS Schema Conversion Tool moteur a été amélioré afin d'offrir de meilleures conversions pour les migrations hétérogènes.	24 juin 2017
1.0.603	Les agents d'extraction de données prennent en charge les filtres	Vous pouvez désormais filtrer les données extraites par les agents des instructions à partir de votre entrepôt de données. Pour de plus amples informations, veuillez consulter <a href="#">Création de règles de migration de données dans AWS SCT</a> .	16 juin 2017

Version	Modification	Description	Date de modification
1.0.603	AWS SCT prend en charge des versions d'entrepôt de données supplémentaires	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour convertir vos schémas Teradata 13 et Oracle Data Warehouse 10 en schémas Amazon Redshift équivalents. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT</a> .	16 juin 2017
1.0.602	Les agents d'extraction de données prennent en charge des entrepôts de données supplémentaires	Vous pouvez désormais utiliser des agents d'extraction de données pour extraire des données de vos entrepôts de données Microsoft SQL Server. Pour de plus amples informations, veuillez consulter <a href="#">Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool</a> .	11 mai 2017
1.0.602	Les agents d'extraction de données peuvent copier des données vers Amazon Redshift	Les agents d'extraction de données disposent désormais de trois modes de chargement. Vous pouvez désormais spécifier si vous souhaitez simplement extraire vos données, les extraire et simplement les charger sur Amazon S3, ou les extraire, charger et copier vos données directement dans Amazon Redshift. Pour de plus amples informations, veuillez consulter <a href="#">Création, exécution et surveillance d'une tâche d'extraction de AWS SCT données</a> .	11 mai 2017

Version	Modification	Description	Date de modification
1.0.601	AWS SCT prend en charge des entrepôts de données supplémentaires	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour convertir vos schémas Vertica et Microsoft SQL Server en schémas Amazon Redshift équivalents. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT</a> .	18 avril 2017
1.0.601	Les agents d'extraction de données prennent en charge des entrepôts de données supplémentaires	Vous pouvez désormais utiliser des agents d'extraction de données pour extraire des données de vos entrepôts de données Greenplum, Netezza, et Vertica. Pour de plus amples informations, veuillez consulter <a href="#">Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool</a> .	18 avril 2017
1.0.601	Les agents d'extraction de données prennent en charge des systèmes d'exploitation supplémentaires	Vous pouvez désormais installer des agents d'extraction de données sur des ordinateurs exécutant des systèmes d'exploitation macOS et Microsoft Windows. Pour de plus amples informations, veuillez consulter <a href="#">Installation d'agents d'extraction</a> .	18 avril 2017

Version	Modification	Description	Date de modification
1.0.601	Les agents d'extraction de données se chargent automatiquement sur Amazon S3	Les agents d'extraction de données chargent désormais automatiquement vos données extraites sur Amazon S3. Pour de plus amples informations, veuillez consulter <a href="#">Sortie de la tâche d'extraction de données</a> .	18 avril 2017
1.0.600	Agents d'extraction de données	Vous pouvez désormais installer des agents d'extraction de données qui extraient les données de votre entrepôt de données et les préparent pour une utilisation avec Amazon Redshift. Vous pouvez utiliser le AWS Schema Conversion Tool pour enregistrer les agents et créer des tâches d'extraction de données pour eux. Pour de plus amples informations, veuillez consulter <a href="#">Migration des données de l'entrepôt de données sur site vers Amazon Redshift avec AWS Schema Conversion Tool</a> .	16 février 2017
1.0.600	Retours des clients	Vous pouvez maintenant fournir des commentaires sur AWS Schema Conversion Tool. Vous pouvez générer un rapport de bogue, soumettre une demande de fonctionnalité ou fournir des informations générales. Pour de plus amples informations, veuillez consulter <a href="#">Fournir des commentaires</a> .	16 février 2017

Version	Modification	Description	Date de modification
1.0.502	Intégration avec AWS DMS	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour créer des AWS DMS points de terminaison et des tâches. Vous pouvez exécuter et surveiller les tâches depuis AWS SCT. Pour de plus amples informations, veuillez consulter <a href="#">Intégration AWS Database Migration Service avec AWS Schema Conversion Tool</a> .	20 décembre 2016
1.0.502	Amazon Aurora avec compatibilité PostgreSQL en tant que base de données cible	AWS Schema Conversion Tool prend désormais en charge Amazon Aurora avec compatibilité PostgreSQL en tant que base de données cible. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas de base de données dans AWS Schema Conversion Tool</a> .	20 décembre 2016
1.0.502	Prise en charge des profils	Vous pouvez désormais stocker plusieurs profils dans AWS Schema Conversion Tool et basculer aisément entre eux. Pour de plus amples informations, veuillez consulter <a href="#">Gestion des profils dans le AWS Schema Conversion Tool</a> .	20 décembre 2016
1.0.501	Prise en charge de Greenplum Database et Netezza	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour convertir vos schémas d'entrepôt de données de Greenplum Database et Netezza vers Amazon Redshift. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT</a> .	17 novembre 2016

Version	Modification	Description	Date de modification
1.0.501	Optimisation de Redshift	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour optimiser vos bases de données Amazon Redshift. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de données depuis Amazon Redshift à l'aide de AWS Schema Conversion Tool</a> .	17 novembre 2016
1.0.500	Règles de mappage	Avant de convertir votre schéma avec le AWS Schema Conversion Tool, vous pouvez désormais définir des règles qui modifient le type de données des colonnes, déplacent des objets d'un schéma à un autre et modifient le nom des objets. Pour de plus amples informations, veuillez consulter <a href="#">Création de règles de migration dans AWS SCT</a> .	4 octobre 2016
1.0.500	Déplacer vers le cloud	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour copier votre schéma de base de données sur site existant vers une instance de base de données Amazon RDS exécutant le même moteur. Cette fonction vous permet d'analyser les économies de coûts potentielles en cas de déplacement vers le cloud et de changement de votre type de licence. Pour de plus amples informations, veuillez consulter <a href="#">À l'aide du rapport d'évaluation figurant dans le AWS Schema Conversion Tool</a> .	4 octobre 2016



Version	Modification	Description	Date de modification
1.0.400	Conversions de schémas d'entrepôt de données	Vous pouvez désormais utiliser le AWS Schema Conversion Tool pour convertir vos schémas d'entrepôt de données d'Oracle et Teradata vers Amazon Redshift. Pour de plus amples informations, veuillez consulter <a href="#">Conversion de schémas d'entrepôt de données vers Amazon RDS à l'aide de AWS SCT</a> .	13 juillet 2016
1.0.400	Conversions de code SQL d'application	Vous pouvez désormais utiliser AWS Schema Conversion Tool pour convertir le code SQL dans votre code d'application C++, C#, Java ou autre. Pour de plus amples informations, veuillez consulter <a href="#">Conversion du code SQL d'une application en utilisant AWS SCT</a> .	13 juillet 2016
1.0.400	Nouvelle fonctionnalité	Il contient AWS Schema Conversion Tool désormais un pack d'extension et un assistant pour vous aider à installer, créer et configurer des AWS Lambda fonctions et des bibliothèques Python pour fournir des fonctionnalités de messagerie, de planification des tâches et d'autres fonctionnalités. Pour plus d'informations, consultez <a href="#">Utilisation des AWS Lambda fonctions du pack d' AWS SCT extension</a> et <a href="#">Utilisation de bibliothèques personnalisées pour les packs d' AWS SCT extension</a> .	13 juillet 2016

Version	Modification	Description	Date de modification
1.0.301	SSL Support	Vous pouvez désormais utiliser le protocole SSL (Secure Sockets Layer) pour vous connecter à votre base de données source lorsque vous utilisez AWS Schema Conversion Tool.	19 mai 2016
1.0.203	Nouvelle fonctionnalité	Ajoute une prise en charge de MySQL et PostgreSQL en tant que bases de données source pour les conversions.	11 avril 2016
1.0.202	Publication de maintenance	Ajoute une prise en charge de la modification du code SQL converti qui a été généré pour le moteur de base de données cible. Ajoute des capacités de sélection améliorées dans les arborescences de la base de données source et de l'instance DB cible. Ajoute une prise en charge de la connexion à une base de données source Oracle en utilisant des noms TNS (Transparent Network Substrate).	2 mars 2016
1.0.200	Publication de maintenance	Ajoute une prise en charge de PostgreSQL en tant que moteur de base de données cible. Ajoute la possibilité de générer un schéma converti sous forme de scripts et d'enregistrer les scripts dans des fichiers avant d'appliquer le schéma à l'instance DB cible.	14 janvier 2016
1.0.103	Publication de maintenance	Ajoute une fonctionnalité de projet hors connexion, la possibilité de rechercher les nouvelles versions et la gestion de la mémoire et des performances.	2 décembre 2015

Version	Modification	Description	Date de modification
1.0.101	Publication de maintenance	Ajoute l'Assistant Create New Database Migration Project. Ajoute la possibilité d'enregistrer le rapport d'évaluation de migration de base de données sous forme de fichier PDF.	19 octobre 2015
1.0.100	Version préliminaire	Fournit le guide de l'utilisateur pour la version AWS Schema Conversion Tool préliminaire.	7 octobre 2015

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.