aws

Leitfaden

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: Leitfaden

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS SCT ... 1
Übersicht über die Schemakonvertierung ... 5
Feedback geben ... 7
Installation und Konfiguration AWS SCT ... 8
Wird installiert AWS SCT ... 8
Die Installation wird validiert ... 10
Überprüfung der Prüfsumme der Datei AWS SCT ... 10
Überprüfung der AWS SCT RPM-Dateien auf Fedora ... 11
Überprüfung der AWS SCT DEB-Dateien auf Ubuntu ... 12
Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows ... 12
Installation von JDBC-Treibern ... 13
Installation von JDBC-Treibern unter Linux ... 16
Speichern von Treiberpfaden in den globalen Einstellungen ... 17
Aktualisierung AWS SCT ... 18
AWS Schema Conversion Tool CLI ... 19
AWS SCT Benutzerschnittstelle ... 20
Projektfenster ... 20
Projekte starten und verwalten ... 22
Verwendung des Assistenten ... 23
Projekte speichern ... 27
Hinzufügen von Datenbankservern ... 28
Offlinemodus ... 29
Baumfilter ... 30
... 31
Eine Dateiliste für den Baumfilter importieren ... 33
Schemas ausblenden ... 34
Bewertungsbericht ... 36
Schemas werden konvertiert ... 40
Anwenden der konvertierten Schemas ... 43
Profile verwalten ... 44
AWS Anmeldeinformationen werden gespeichert ... 45
Das Standardprofil für ein Projekt festlegen ... 47
Berechtigungen für die Verwendung des AWS Dienstprofils ... 47
Secrets Manager konfigurieren ... 48

Speichern von Datenbankkennwörtern ... 49
Ansicht UNION ALL ... 49
Tastenkombinationen ... 50
Erste Schritte ... 52
Verbindung zu Quelldatenbanken herstellen ... 54
Verbindung zu verschlüsseltem Amazon RDS und Aurora herstellen ... 55
Verbindung zu Apache Cassandra herstellen ... 58
Verbindung zu Apache Cassandra als Quelle herstellen ... 58
Verbindung zu Apache Hadoop herstellen ... 60
Voraussetzungen für die Verwendung von Apache Hadoop als Quelle ... 61
Berechtigungen für Hive als Quelle ... 62
Berechtigungen für HDFS als Quelle ... 63
Berechtigungen für HDFS als Ziel ... 63
Verbindung zu Apache Hadoop als Quelle herstellen ... 63
Verbindung zu Hive und HDFS herstellen ... 65
Verbindung zu Amazon EMR als Ziel herstellen ... 69
Verbindung zu Apache Oozie herstellen ... 72
Voraussetzungen ... 72
Verbindung zu Apache Oozie als Quelle herstellen ... 73
Berechtigungen für AWS Lambda ... 74
AWS Step Functions Als Ziel wird eine Verbindung hergestellt ... 77
Verbindung zu Azure SQL herstellen ... 78
Rechte für Azure SQL Database ... 78
Verbindung zur Azure SQL-Datenbank als Quelle herstellen ... 79
Verbindung zu IBM DB2 für z/OS herstellen ... 81
Voraussetzungen für Db2 for z/OS ... 81
Berechtigungen für Db2 for z/OS ... 81
Verbindung zu Db2 for z/OS als Quelle herstellen ... 83
Berechtigungen für MySQL als Ziel ... 85
Berechtigungen für PostgreSQL als Ziel ... 87
Einstellungen für die Konvertierung von Db2 for z/OS nach PostgreSQL ... 87
IBM Db2 LUW-Datenbanken ... 89
Rechte für Db2 LUW ... 90
Verbindung zu Db2 LUW als Quelle herstellen ... 92
Db2 LUW nach PostgreSQL ... 95
Db2 LUW nach MySQL ... 97

Verwenden von MySQL als Quelle ... 99
Berechtigungen für MySQL ... 99
Verbindung zu MySQL als Quelle herstellen ... 99
Berechtigungen für PostgreSQL als Ziel ... 102
Oracle-Datenbanken ... 103
Berechtigungen für Oracle ... 104
Verbindung zu Oracle als Quelle herstellen ... 104
Oracle zu PostgreSQL ... 110
Oracle in MySQL ... 116
Oracle zu Amazon RDS for Oracle ... 126
PostgreSQL-Datenbanken ... 133
Berechtigungen für PostgreSQL ... 133
Verbindung zu PostgreSQL als Quelle herstellen ... 134
Berechtigungen für MySQL als Ziel ... 137
SAP-Datenbanken ... 138
Privilegien für SAP ASE ... 139
Verbindung zu SAP ASE als Quelle herstellen ... 140
Berechtigungen für MySQL als Ziel ... 142
Einstellungen für die Konvertierung von SAP ASE nach MySQL ... 143
Berechtigungen für PostgreSQL als Ziel ... 144
Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL ... 145
SQL Server-Datenbanken ... 146
Berechtigungen für Microsoft SQL Server ... 147
Verwenden der Windows-Authentifizierung mit Microsoft SQL Server ... 148
Verbindung zu SQL Server als Quelle herstellen ... 151
SQL Server in MySQL ... 154
SQL Server in PostgreSQL ... 158
SQL Server in Amazon RDS SQL Server ... 195
Data Warehouses ... 196
Amazon Redshift ... 197
Azure Synapse Analytics als Quelle ... 203
BigQuery als Quelle ... 209
Greenplum-Datenbanken ... 215
Netezza-Datenbanken ... 222
Oracle Data Warehouse ... 232
Snowflake ... 240

SQL Server-Data Warehouses ... 249
Teradata-Datenbanken ... 256
Vertica-Datenbanken ... 272
Zuordnung von Datentypen ... 280
Neue Datentyp-Zuordnung ... 281
Bearbeiten von Datentypzuordnungen ... 281
Zuordnung virtueller Ziele ... 283
Einschränkungen bei der Zuordnung von Datentypen ... 284
Berichte ... 285
Bewertungsbericht ... 285
Bewertungsbericht erstellen ... 286
Bewertungsbericht anzeigen ... 287
Speichern des Bewertungsberichts ... 292
Konfiguration eines Bewertungsberichts ... 295
Bewertungsbericht für mehrere Server ... 298
Schemas werden konvertiert ... 308
Anwendung von Migrationsregeln ... 310
Migrationsregeln erstellen ... 311
Migrationsregeln exportieren ... 313
Manuelles Konvertieren von Schemas ... 313
Konvertieren des Schemas ... 314
Konvertiertes Schema bearbeiten ... 317
Löschen eines konvertierten Schemas ... 318
Manuelles Konvertieren von Schemas ... 318
Ihr Quellschema ändern ... 318
Ihr Zielschema ändern ... 318
Aktualisierung und Aktualisierung von Schemas ... 319
Konvertierte Schemas speichern und anwenden ... 320
Speichern des konvertierten Schemas ... 320
Ihr konvertiertes Schema wird angewendet ... 321
Das Erweiterungspaket-Schema ... 322
Schemas vergleichen ... 322
Verwandte transformierte Objekte anzeigen ... 324
Data Warehouse-Schemas konvertieren ... 325
Berechtigungen für Amazon Redshift ... 327
Auswahl von Optimierungsstrategien und -regeln ... 328

Statistiken sammeln oder hochladen ... 330
Migrationsregeln erstellen ... 331
Migrationsregeln erstellen ... 332
Migrationsregeln exportieren ... 334
Konvertieren Sie Ihr Schema ... 334
Konvertieren des Schemas ... 335
Konvertiertes Schema bearbeiten ... 338
Löschen eines konvertierten Schemas ... 339
Schlüssel verwalten und anpassen ... 339
Verwandte Themen ... 340
Erstellung und Verwendung des Bewertungsberichts ... 340
Erstellen eines Berichts zur Bewertung der Datenbankmigration ... 340
Übersicht ... 341
Aktionselemente ... 343
Speichern des Bewertungsberichts ... 343
Umgang mit manuellen Konvertierungen ... 344
Ändern Sie Ihr Quellschema ... 345
Ändern Sie Ihr Zielschema ... 345
Aktualisierung und Aktualisierung Ihres konvertierten Schemas ... 346
Speichern und Anwenden Ihres konvertierten Schemas ... 346
Speichern Sie Ihr konvertiertes Schema in einer Datei ... 347
Ihr konvertiertes Schema wird angewendet ... 347
Das Erweiterungspaket-Schema ... 348
Python-Bibliotheken ... 349
Amazon Redshift Redshift-Daten konvertieren ... 349
Optimieren Sie Ihre Amazon Redshift Redshift-Datenbank ... 349
Daten mithilfe von ETL konvertieren ... 352
ETL-Prozesse ... 353
Voraussetzungen ... 354
AWS Glue Datenkatalog ... 355
Einschränkungen ... 355
Schritt 1: Erstellen eines neuen -Projekts ... 357
Schritt 2: Erstellen Sie einen AWS Glue Job ... 358
ETL-Prozesse mit Python ... 360
Schritt 1: Erstellen einer Datenbank ... 360
Schritt 2: Stellen Sie eine Verbindung her ... 361

Schritt 3: Erstellen Sie einen AWS Glue Crawler ... 362
Informatica ETL-Skripts ... 365
SSIS-Pakete ... 370
Unterstützte SSIS-Komponenten ... 374
SSIS zu AWS Glue Studio ... 376
Voraussetzungen ... 376
Hinzufügen von SSIS-Paketen zu Ihrem Projekt AWS SCT ... 378
SSIS-Pakete werden konvertiert ... 379
AWS Glue Studio Jobs erstellen ... 380
Erstellen eines Bewertungsberichts zur SSIS-Konvertierung ... 382
Unterstützte SSIS-Komponenten ... 383
Teradata BTEQ-Skripts ... 384
Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT ... 386
Konfiguration von Substitutionsvariablen in BTEQ-Skripten ... 387
BTEQ-Skripts konvertieren ... 388
Verwaltung von BTEQ-Skripten ... 388
Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung ... 389
Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte ... 390
Shell-Skripts ... 390
Hinzufügen von Shell-Skripten zu Ihrem Projekt AWS SCT ... 391
Konfiguration von Substitutionsvariablen in Shell-Skripten ... 392
Shell-Skripte konvertieren ... 393
Verwaltung von Shell-Skripten ... 394
Erstellen eines Bewertungsberichts zur Shell-Skriptkonvertierung ... 395
Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte ... 396
FastExport Skripte ... 396
Hinzufügen von FastExport Jobskripten zu Ihrem AWS SCT Projekt ... 397
Konfiguration von Substitutionsvariablen in Jobskripten FastExport ... 398
Jobskripte konvertieren FastExport ... 399
Jobskripte verwalten FastExport ... 400
Erstellen eines Bewertungsberichts zur Konvertierung von FastExport Jobskripten ... 400
Bearbeiten und Speichern Ihrer konvertierten FastExport Jobskripte ... 401
FastLoad Skripte ... 402
Hinzufügen von FastLoad Jobskripten zu Ihrem AWS SCT Projekt ... 403
Konfiguration von Substitutionsvariablen in Jobskripten FastLoad ... 404
Jobskripte konvertieren FastLoad ... 405

Jobskripte verwalten FastLoad ... 406
Erstellen eines Bewertungsberichts zur Konvertierung von FastLoad Jobskripten ... 407
Bearbeiten und Speichern Ihrer konvertierten FastLoad Jobskripte ... 408
MultiLoad Skripte ... 408
Hinzufügen von MultiLoad Jobskripten zu Ihrem AWS SCT Projekt ... 409
Konfiguration von Substitutionsvariablen in Jobskripten MultiLoad ... 410
Jobskripte konvertieren MultiLoad ... 411
Jobskripte verwalten MultiLoad ... 412
Erstellen eines Bewertungsberichts zur Konvertierung von MultiLoad Jobskripten ... 413
Bearbeiten und Speichern Ihrer konvertierten MultiLoad Jobskripte ... 414
Migration von Big-Data-Frameworks ... 415
Migration von Hadoop-Workloads ... 415
Übersicht ... 416
Schritt 1: Connect zu Ihren Hadoop-Clustern her ... 417
Schritt 2: Richten Sie die Zuordnungsregeln ein ... 417
Schritt 3: Erstellen Sie einen Bewertungsbericht ... 419
Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR ... 421
Ausführen Ihres CLI-Skripts ... 422
Verwaltung Ihres Migrationsprojekts ... 422
Konvertierung von Oozie-Workflows; ... 424
Übersicht ... 424
Schritt 1: Connect zu Ihren Quell- und Zieldiensten her ... 426
Schritt 2: Richten Sie die Zuordnungsregeln ein ... 427
Schritt 3: Parameter konfigurieren ... 427
Schritt 4: Erstellen eines Bewertungsberichts ... 429
Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows in AWS Step Functions ... 431
Ausführen Ihres CLI-Skripts ... 434
Unterstützte Knoten ... 434
Integrieren mit AWS DMS ... 436
Verwenden eines AWS SCT Replikationsagenten mit AWS DMS ... 436
Verwenden eines AWS SCT Datenextraktionsagenten mit AWS DMS ... 436
Erhöhung der Protokollierungsebenen bei Verwendung AWS SCT mit AWS DMS ... 437
Migration aus einem Data Warehouse ... 439
Voraussetzungen ... 442
Amazon S3 S3-Einstellungen ... 442
Übernahme von IAM-Rollen ... 443

Sicherheitseinstellungen ... 445
Konfigurationseinstellungen ... 446
Installieren von -Agenten ... 446
Agenten konfigurieren ... 448
Installation und Konfiguration spezieller Kopieragenten ... 450
Agenten starten ... 452
Registrierung von Agenten ... 453
Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen ... 454
Regeln für die Datenmigration erstellen ... 456
Ändern der Extraktor- und Kopiereinstellungen für die Datenmigration ... 457
Daten sortieren ... 460
Eine AWS SCT Aufgabe erstellen, ausführen und überwachen ... 462
Exportieren und Importieren einer Datenextraktionsaufgabe ... 466
Datenextraktion mit einem AWS Snowball Edge Edge-Gerät ... 467
Step-by-step Verfahren für die Migration von Daten mithilfe von Edge AWS SCT AWS Snowball Edge ... 468
Ausgabe der Datenextraktionsaufgabe ... 471
Verwenden von virtueller Partitionierung ... 473
Einschränkungen bei der Erstellung einer virtuellen Partitionierung ... 473
RANGE-Partitionstyp ... 474
LIST-Partitionstyp ... 475
DATE AUTO SPLIT Partitionstyp ... 476
Verwenden der nativen Partitionierung ... 477
Arbeitet mit LOBs ... 479
Bewährte Methoden und Problembehebung ... 480
Anwendungs-SQL konvertieren ... 482
Überblick über die Konvertierung von Anwendungs-SQL ... 482
SQL-Code ... 483
Projekte zur Konvertierung generischer Anwendungen erstellen ... 483
Verwaltung von Projekten zur Anwendungskonvertierung ... 488
Analysieren und konvertieren Sie Ihren SQL-Code ... 489
Erstellung und Verwendung des Bewertungsberichts ... 490
Bearbeiten und Speichern Ihres konvertierten SQL-Codes ... 491
SQL-Code in C#-Anwendungen ... 492
Projekte zur Konvertierung von C#-Anwendungen erstellen ... 492
Konvertieren Sie den SQL-Code Ihrer C#-Anwendung ... 494

Speichern Sie Ihren konvertierten Anwendungscode ... 495
Verwaltung von Projekten zur Konvertierung von C#-Anwendungen ... 495
Erstellen eines Bewertungsberichts zur C#-Anwendungskonvertierung ... 496
SQL-Code in C++ ... 498
Projekte zur Konvertierung von C++-Anwendungen erstellen ... 498
Konvertieren Sie den SQL-Code Ihrer C++-Anwendung ... 500
Speichern Sie Ihren konvertierten Anwendungscode ... 501
Verwaltung von Projekten zur Konvertierung von C++-Anwendungen ... 502
Erstellen eines Bewertungsberichts zur Konvertierung von C++-Anwendungen ... 504
SQL-Code in Java ... 505
Projekte zur Konvertierung von Java-Anwendungen erstellen ... 505
Konvertieren Sie den SQL-Code Ihrer Java-Anwendung ... 507
Speichern Sie Ihren konvertierten Anwendungscode ... 509
Verwaltung von Projekten zur Konvertierung von Java-Anwendungen ... 509
Erstellen eines Bewertungsberichts zur Konvertierung von Java-Anwendungen ... 510
SQL-Code in Pro*C ... 512
Projekte zur Konvertierung von Pro*C-Anwendungen erstellen ... 512
Konvertieren Sie den SQL-Code Ihrer Pro*C-Anwendung ... 514
Bearbeiten und Speichern Ihres konvertierten Anwendungscodes ... 515
Verwaltung von Projekten zur Konvertierung von Pro*C-Anwendungen ... 516
Erstellen eines Bewertungsberichts zur Konvertierung von Pro*C-Anwendungen ... 517
Erweiterungspakete ... 520
Berechtigungen für die Verwendung des Erweiterungspakets ... 522
Verwenden des Erweiterungspaket-Schemas ... 523
Benutzerdefinierte Bibliotheken für Erweiterungspakete ... 524
Anwenden des Erweiterungspakets ... 524
Verwenden der Lambda-Funktionen aus dem AWS SCT Erweiterungspaket ... 527
Verwendung von AWS Lambda Funktionen zur Emulierung von Datenbankfunktionen ... 527
Anwenden des Erweiterungspakets zur Unterstützung von Lambda-Funktionen ... 527
Konfiguration der Funktionen des Erweiterungspakets ... 529
Bewährte Methoden ... 531
Konfiguration von zusätzlichem Speicher ... 531
Standard-Projektordner ... 531
Erhöhung der Geschwindigkeit der Datenmigration ... 532
Zunehmende Protokollierungsinformationen ... 532
Fehlerbehebung ... 536

Objekte können nicht aus einer Oracle-Quelldatenbank geladen werden. ... 536
Warnmeldung ... 536
CLI Reference ... 538
Voraussetzungen ... 538
interaktiver Modus ... 538
Beispiele ... 540
CLI-Szenarien abrufen ... 540
Beispiele ... 545
CLI-Szenarien bearbeiten ... 545
Skriptmodus ... 547
Beispiele ... 547
Referenzmaterial ... 548
Versionshinweise ... 549
Versionshinweise — 676 ... 549
Versionshinweise — 675 ... 554
Versionshinweise — 674 ... 557
Versionshinweise — 673 ... 564
Versionshinweise — 672 ... 569
Versionshinweise — 671 ... 578
Versionshinweise — 670 ... 587
Versionshinweise — 669 ... 592
Versionshinweise — 668 ... 597
Versionshinweise — 667 ... 604
Versionshinweise — 666 ... 608
Versionshinweise — 665 ... 612
Versionshinweise — 664 ... 616
Versionshinweise — 663 ... 619
Versionshinweise — 662 ... 622
Versionshinweise — 661 ... 627
Versionshinweise — 660 ... 632
Versionshinweise — 659 ... 636
Versionshinweise — 658 ... 640
Versionshinweise — 657 ... 645
Versionshinweise — 656 ... 650
Versionshinweise — 655 ... 653
Versionshinweise — 654 ... 656

Versionshinweise — 653 660
Versionshinweise — 652 662
Versionshinweise — 651 664
Versionshinweise — 650 666
Versionshinweise — 649 668
Versionshinweise — 648 671
Versionshinweise — 647 673
Versionshinweise — 646 675
Versionshinweise — 645 677
Versionshinweise — 644 678
Versionshinweise — 642 680
Versionshinweise — 641 682
Versionshinweise — 640 683
Version 1.0.640 Oracle-Änderungen 683
Version 1.0.640 Änderungen an Microsoft SQL Server 690
Änderungen an MySQL-Version 1.0.640 695
Version 1.0.640 PostgreSQL-Änderungen 696
Änderungen an der Db2-LUW-Version 1.0.640 699
Teradata-Änderungen an Version 1.0.640 699
Version 1.0.640 ändert sich für andere Engines 701
Dokumentverlauf 704
Frühere Aktualisierungen 721
..... dccxxxi

Was ist der AWS Schema Conversion Tool?

Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um Ihr vorhandenes Datenbankschema von einer Datenbank-Engine in eine andere zu konvertieren. Sie können relationale OLTP-Schemata sowie Data Warehouse-Schemata konvertieren. Ihr konvertiertes Schema ist für Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, PostgreSQL DB, einen Amazon-Aurora-DB-Cluster oder einen Amazon-Redshift-Cluster geeignet. Das konvertierte Schema kann auch mit einer Datenbank auf einer EC2 Amazon-Instance verwendet oder als Daten in einem Amazon S3-Bucket gespeichert werden.

AWS SCT unterstützt mehrere Industriestandards, darunter Federal Information Processing Standards (FIPS), für Verbindungen zu einem Amazon S3 S3-Bucket oder einer anderen AWS Ressource. AWS SCT entspricht auch dem Federal Risk and Authorization Management Program (FedRAMP). Einzelheiten zu den Compliance-Maßnahmen AWS und den Bemühungen zur Einhaltung der Vorschriften finden Sie unter AWS Leistungen nach Compliance-Programmen.

AWS SCT unterstützt die folgenden OLTP-Konvertierungen.

Quelldatenbank	Zieldatenbank
IBM Db2 für z/OS (Version 12)	Amazon Aurora MySQL-kompatible Edition (Aurora MySQL), Amazon Aurora PostgreSQL-kompatible Edition (Aurora PostgreSQL), MySQL, PostgreSQL Weitere Informationen finden Sie unter Verbindung zu IBM DB2 für z/OS herstellen.
IBM Db2 LUW (Versionen 9.1, 9.5, 9.7, 10.5, 11.1 und 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Weitere Informationen finden Sie unter IBM Db2 LUW-Datenbanken.
Microsoft Azure SQL Database	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Quelldatenbank	Zieldatenbank
	Weitere Informationen finden Sie unter Verbindung zu Azure SQL herstellen.
Microsoft SQL Server (Version 2008 R2, 2012, 2014, 2016, 2017, 2019 und 2022)	Aurora MySQL, Aurora PostgreSQL, Babelfish für Aurora PostgreSQL (nur für Bewertung sberichte), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL Weitere Informationen finden Sie unter SQL Server-Datenbanken.
MySQL (Version 5.5 und höher)	Aurora PostgreSQL, MySQL, PostgreSQL Weitere Informationen finden Sie unter Verwenden von MySQL als Quelle. Sie können Schema und Daten von MySQL zu einem Aurora MySQL-DB-Cluster migrieren , ohne es zu verwenden AWS SCT. Weitere Informationen finden Sie unter Daten in einen Amazon Aurora Aurora-DB-Cluster migrieren.
Oracle (Version 10.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL Weitere Informationen finden Sie unter Oracle-Datenbanken.
PostgreSQL (Version 9.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Weitere Informationen finden Sie unter PostgreSQL-Datenbanken.

Quelldatenbank	Zieldatenbank
SAP ASE (Versionen 12.5.4, 15.0.2, 15.5, 15.7 und 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Weitere Informationen finden Sie unter SAP-Datenbanken.

AWS SCT unterstützt die folgenden Data Warehouse-Konvertierungen.

Quell-Data Warehouse	Ziel-Data-Warehouse
Amazon Redshift	Amazon Redshift Weitere Informationen finden Sie unter Amazon Redshift.
Azure Synapse Analytics	Amazon Redshift Weitere Informationen finden Sie unter Azure Synapse Analytics als Quelle.
BigQuery	Amazon Redshift Weitere Informationen finden Sie unter BigQuery als Quelle.
Greenplum-Datenbank (Versionen 4.3 und 6.21)	Amazon Redshift Weitere Informationen finden Sie unter Greenplum-Datenbanken.
Microsoft SQL Server (Version 2008 und höher)	Amazon Redshift Weitere Informationen finden Sie unter SQL Server-Data Warehouses.
Netezza (Version 7.0.3 und höher)	Amazon Redshift

Quell-Data Warehouse	Ziel-Data-Warehouse
	Weitere Informationen finden Sie unter Netezza-Datenbanken.
Oracle (Version 10.1 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Oracle Data Warehouse.
Schneeflocke (Version 3)	Amazon Redshift Weitere Informationen finden Sie unter Snowflake.
Teradata (Version 13 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Teradata-Datenbanken.
Vertica (Version 7.2.2 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Vertica-Datenbanken.

AWS SCT unterstützt die folgenden Daten-NoSQL-Datenbankkonvertierungen.

Quelldatenbank	Zieldatenbank
Apache Cassandra (Versionen 2.1.x, 2.2.16 und 3.11.x)	Amazon-DynamoDB Weitere Informationen finden Sie unter Verbindung zu Apache Cassandra herstellen.

AWS SCT unterstützt Konvertierungen der folgenden ETL-Prozesse (Extrahieren, Transformieren und Laden). Weitere Informationen finden Sie unter Daten mithilfe von ETL konvertieren.

Quelle	Ziel
ETL-Skripts von Informatica	Informatica
ETL-Pakete für Microsoft SQL Server Integration Services (SSIS)	AWS Glue oder AWS Glue Studio
Shell-Skripts mit eingebetteten Befehlen aus Teradata Basic Teradata Query (BTEQ)	Amazon-Redshift-RSQL
Teradata BTEQ ETL-Skripts	AWS Glue oder Amazon Redshift RSQL
FastExport Teradata-Jobskripte	Amazon-Redshift-RSQL
FastLoad Teradata-Jobskripte	Amazon-Redshift-RSQL
MultiLoad Teradata-Jobskripte	Amazon-Redshift-RSQL

AWS SCT unterstützt die folgenden Big-Data-Framework-Migrationen. Weitere Informationen finden Sie unter Migration von Big-Data-Frameworks.

Quelle	Ziel
Apache Hive (Version 0.13.0 und höher)	Hive auf Amazon EMR
Apache HDFS	Amazon S3 oder HDFS auf Amazon EMR
Apache Oozie	AWS Step Functions

Übersicht über die Schemakonvertierung

AWS SCT bietet eine projektbasierte Benutzeroberfläche, um das Datenbankschema Ihrer Quelldatenbank automatisch in ein Format zu konvertieren, das mit Ihrer Amazon RDS-Zielinstanz kompatibel ist. Falls das Schema aus Ihrer Quelldatenbank nicht automatisch konvertiert werden kann, finden Sie AWS SCT hier eine Anleitung, wie Sie ein entsprechendes Schema in Ihrer Amazon RDS-Zieldatenbank erstellen können.

Informationen zur Installation finden Sie AWS SCT unterInstallation und Konfiguration AWS Schema Conversion Tool.

Eine Einführung in die AWS SCT Benutzeroberfläche finden Sie unterNavigieren in der Benutzeroberfläche des AWS SCT.

Weitere Informationen zum Umwandlungsverfahren finden Sie unter Datenbankschemas konvertieren in AWS Schema Conversion Tool.

Neben der Konvertierung Ihres vorhandenen Datenbankschemas von einer Datenbank-Engine in eine andere AWS SCT bietet es einige zusätzliche Funktionen, mit denen Sie Ihre Daten und Anwendungen in die AWS Cloud verschieben können:

- Sie können Datenextraktionsagenten verwenden, um Daten aus Ihrem Data Warehouse zu extrahieren, um sie auf die Migration zu Amazon Redshift vorzubereiten. Zur Verwaltung der Agenten für die Datenextraktion können Sie das AWS SCT verwenden. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.
- Sie können sie verwenden AWS SCT , um AWS DMS Endpunkte und Aufgaben zu erstellen. Sie können diese Aufgaben von AWS SCT ausführen und überwachen. Weitere Informationen finden Sie unter Integrieren AWS Database Migration Service mit AWS Schema Conversion Tool.
- In einigen Fällen können Datenbankfunktionen nicht in äquivalente Amazon RDS- oder Amazon Redshift Redshift-Funktionen konvertiert werden. Der AWS SCT Erweiterungspaket-Assistent kann Ihnen bei der Installation von AWS Lambda Funktionen und Python-Bibliotheken helfen, um die Funktionen zu emulieren, die nicht konvertiert werden können. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.
- Sie können es verwenden AWS SCT , um Ihre bestehende Amazon Redshift Redshift-Datenbank zu optimieren. AWS SCT empfiehlt Sortierschlüssel und Verteilungsschlüssel zur Optimierung Ihrer Datenbank. Weitere Informationen finden Sie unter Daten aus Amazon Redshift konvertieren mit AWS Schema Conversion Tool.
- Sie können Ihr vorhandenes AWS SCT lokales Datenbankschema in eine Amazon RDS-DB-Instance kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind.
- Sie können mit dem AWS SCT SQL-Code in Ihren C++-, C #-, Java- oder anderen Anwendungscode konvertieren. Sie können den konvertierten SQL-Code anzeigen, analysieren,

bearbeiten und speichern. Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren mit AWS SCT.

- Sie können AWS SCT es zur Migration von Extraktions-, Transformations- und Ladeprozessen (ETL) verwenden. Weitere Informationen finden Sie unter Konvertieren von Daten mithilfe von ETL-Prozessen in AWS Schema Conversion Tool.

Feedback geben

Sie können Feedback geben zu AWS SCT. Sie können einen Fehlerbericht einreichen, eine Funktionsanfrage stellen oder allgemeine Informationen bereitstellen.

Um Feedback zu geben AWS SCT

1. Starte das AWS Schema Conversion Tool.
2. Öffnen Sie das Menü Help und klicken Sie auf Leave Feedback. Das Dialogfeld Leave Feedback wird angezeigt.
3. Wählen Sie unter Area eine der Optionen Information, Bug report oder Feature request aus.
4. Wählen Sie für Source database Ihre Quelldatenbank aus. Wählen Sie Any aus, wenn Ihr Feedback sich nicht auf eine bestimmte Datenbank bezieht.
5. Wählen Sie für Target database Ihre Zieldatenbank aus. Wählen Sie Any aus, wenn Ihr Feedback sich nicht auf eine bestimmte Datenbank bezieht.
6. Geben Sie unter Title einen Titel für Ihr Feedback ein.
7. Geben Sie unter Message Ihr Feedback ein.
8. Klicken Sie auf Send, um das Feedback abzugeben.

Installation und Konfiguration AWS Schema Conversion Tool

The AWS Schema Conversion Tool (AWS SCT) ist eine eigenständige Anwendung, die eine projektbasierte Benutzeroberfläche bietet. AWS SCT ist für Microsoft Windows, Fedora Linux und Ubuntu Linux verfügbar. AWS SCT wird nur auf 64-Bit-Betriebssystemen unterstützt.

Um sicherzustellen, dass Sie die richtige Version der AWS SCT Distributionsdatei erhalten, führen wir nach dem Herunterladen der komprimierten Datei Schritte zur Überprüfung durch. Sie können die Datei unter Verwendung der bereitgestellten Schritte überprüfen.

AWS SCT ist sowohl als eigenständige Anwendung als auch als Befehlszeilentool verfügbar. Hinweise zum Befehlszeilentool finden Sie unter. AWS Schema Conversion Tool CLI

Themen

- Wird installiert AWS Schema Conversion Tool
- Die Installation wird validiert AWS Schema Conversion Tool
- Installation von JDBC-Treibern für AWS Schema Conversion Tool
- Aktualisierung AWS Schema Conversion Tool
- AWS Schema Conversion Tool CLI

Wird installiert AWS Schema Conversion Tool

Sie können die Installation AWS SCT auf den folgenden Betriebssystemen durchführen:

- Microsoft Windows 10
- Fedora Linux 3.6 und höher
- Ubuntu Linux 18 und höher

Um zu installieren AWS SCT

1. Laden Sie die komprimierte Datei, die das AWS SCT Installationsprogramm enthält, über den Link für Ihr Betriebssystem herunter. Alle komprimierten Dateien haben eine ZIP-Erweiterung. Wenn Sie die AWS SCT Installationsdatei extrahieren, wird sie das für Ihr Betriebssystem geeignete Format haben.

- Microsoft Windows
- Ubuntu Linux (.deb)
- Fedora Linux (.rpm)

2. Extrahieren Sie die AWS SCT Installationsdatei für Ihr Betriebssystem, wie unten gezeigt.

Betriebssystem	Dateiname
Fedora Linux	`aws-schema-conversion-tool-1.0.` *`build-number`* `.x86_64.rpm`
Microsoft Windows	`AWS Schema Conversion Tool-1.0.` *`build-number`* `.msi`
Ubuntu Linux	`aws-schema-conversion-tool-1.0.` *`build-number`* `.deb`

3. Führen Sie die im vorherigen Schritt extrahierte AWS SCT Installationsdatei aus. Befolgen Sie die Anleitung für Ihr Betriebssystem:

Betriebssystem	Anweisungen installieren
Fedora Linux	Führen Sie in dem Verzeichnis, in das Sie die Datei heruntergeladen haben, den folgenden Befehl aus: `sudo yum install aws-schema-conversion-tool-1.0.` *`build-number`* `.x86_64.rpm`
Microsoft Windows	Doppelklicken Sie auf die Datei, um das Installationsprogramm auszuführen.
Ubuntu Linux	Führen Sie in dem Verzeichnis, in das Sie die Datei heruntergeladen haben, den folgenden Befehl aus: `sudo dpkg -i aws-schema-conversion-tool-1.0.` *`build-number`* `.deb`

4. Laden Sie die Java Database Connectivity (JDBC) -Treiber für Ihre Quell- und Zieldatenbank-Engines herunter. Informationen und Download-Links finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

Jetzt haben Sie die Einrichtung der AWS SCT Anwendung abgeschlossen. Doppelklicken Sie auf das Anwendungssymbol, um es auszuführen AWS SCT.

Die Installation wird validiert AWS Schema Conversion Tool

Es gibt mehrere Möglichkeiten, die Distributionsdatei von zu überprüfen AWS SCT. Am einfachsten ist es, die Prüfsumme der Datei mit der veröffentlichten Prüfsumme von zu vergleichen. AWS Als zusätzliche Sicherheitsstufe können Sie mithilfe der folgenden Verfahren die Distributionsdatei basierend auf dem Betriebssystem, auf dem die Datei installiert ist, überprüfen.

In diesem Abschnitt werden folgende Themen behandelt.

Themen

- Überprüfung der Prüfsumme der Datei AWS SCT
- Überprüfung der AWS SCT RPM-Dateien auf Fedora
- Überprüfung der AWS SCT DEB-Dateien auf Ubuntu
- Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows

Überprüfung der Prüfsumme der Datei AWS SCT

Um Fehler zu erkennen, die beim Herunterladen oder Speichern der AWS SCT komprimierten Datei entstanden sein könnten, können Sie die Dateiprüfsumme mit einem Wert vergleichen, der von bereitgestellt wird. AWS AWS verwendet den SHA256 Algorithmus für die Prüfsumme.

Um die AWS SCT Distributionsdatei anhand einer Prüfsumme zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter. Weitere Informationen finden Sie unter Wird installiert AWS Schema Conversion Tool.
2. Laden Sie die neueste Prüfsummendatei mit dem Namen sha256Check.txt herunter. Diese Datei enthält die Prüfsummen für die neueste AWS SCT Version. Die Datei kann beispielsweise wie folgt aussehen:

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

3. Führen Sie den SHA256 Validierungsbefehl für Ihr Betriebssystem in dem Verzeichnis aus, das die Distributionsdatei enthält. Führen Sie beispielsweise den folgenden Befehl unter Linux aus.

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

4. Vergleichen Sie die Ergebnisse des Befehls mit dem Wert in der Datei sha256Check.txt. Wenn die Prüfsummen übereinstimmen, ist es sicher, die Distributionsdatei auszuführen. Wenn die Prüfsummen nicht übereinstimmen, führen Sie die Verteilungsdatei nicht aus und wenden Sie sich an den AWS-Support.

Überprüfung der AWS SCT RPM-Dateien auf Fedora

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Validierungsebene. Alle RPM-Dateien in der Distributionsdatei sind mit einem AWS privaten Schlüssel signiert. Sie finden den öffentlichen GPG-Schlüssel unter amazon.com.public.gpg-key.

Um die AWS SCT RPM-Dateien auf Fedora zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
2. Überprüfen Sie die Prüfsumme der AWS SCT Distributionsdatei.
3. Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die RPM-Datei, die Sie überprüfen möchten.
4. Laden Sie den öffentlichen GPG-Schlüssel unter amazon.com.public.gpg-key herunter.
5. Importieren Sie den öffentlichen Schlüssel in Ihre RPM-DB (stellen Sie sicher, dass Sie über die entsprechenden Berechtigungen verfügen), indem Sie den folgenden Befehl ausführen:

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

6. Überprüfen Sie, ob der Import erfolgreich war, indem Sie den folgenden Befehl ausführen:

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

7. Überprüfen Sie die RPM-Signatur, indem Sie den folgenden Befehl ausführen:

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

Überprüfung der AWS SCT DEB-Dateien auf Ubuntu

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Validierungsebene. Alle DEB-Dateien in der Distributionsdatei werden mit einer von GPG getrennten Signatur signiert.

Um die AWS SCT DEB-Dateien auf Ubuntu zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
2. Überprüfung der Prüfsumme der AWS SCT Distributionsdatei.
3. Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die DEB-Datei, die Sie überprüfen möchten.
4. Laden Sie die abgetrennte Signatur von aws-schema-conversion-tool -1.0.latest.deb.asc herunter.
5. Laden Sie den öffentlichen GPG-Schlüssel unter amazon.com.public.gpg-key herunter.
6. Importieren Sie den öffentlichen GPG-Schlüssel, indem Sie den folgenden Befehl ausführen:

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

7. Überprüfen Sie die Signatur, indem Sie den folgenden Befehl ausführen:

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

Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Überprüfungsebene. Die MSI-Datei hat eine digitale Signatur, mit der Sie überprüfen können, ob sie signiert AWS wurde.

Um die AWS SCT MSI-Datei unter Windows zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
2. Überprüfung der Prüfsumme der AWS SCT Distributionsdatei.
3. Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die MSI-Datei, die Sie überprüfen möchten.
4. Klicken Sie im Windows Explorer mit der rechten Maustaste auf die MSI-Datei und wählen Sie Properties.

5. Wählen Sie die Registerkarte Digital Signatures aus.
6. Überprüfen Sie, ob die digitale Signatur von Amazon Services LLC stammt.

Installation von JDBC-Treibern für AWS Schema Conversion Tool

Laden Sie die JDBC-Treiber für Ihre Quell- und Zieldatenbank-Engines herunter, AWS SCT damit sie ordnungsgemäß funktionieren. Wenn Sie eine virtuelle Zieldatenbankplattform verwenden, müssen Sie den JDBC-Treiber für Ihre Zieldatenbank-Engine nicht herunterladen. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.

Nachdem Sie die Treiber heruntergeladen haben, geben Sie den Speicherort der Treiberdateien an. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

Sie können die Datenbanktreiber an folgenden Speicherorten herunterladen.

> **Important**
>
> Laden Sie die neueste verfügbare Version des Treibers herunter. Die folgende Tabelle enthält die niedrigste Version des Datenbanktreibers, die von unterstützt wird AWS SCT.

Datenbank-Engine	Treiber	Speicherort für den Download
Amazon Aurora MySQL-Compatible Edition	`mysql-connector-java-5.1.6.jar`	https://www.mysql.com/products/Konnektor/
Amazon Aurora PostgreSQL-Compatible Edition	`postgresql-42.2.19.jar`	https://jdbc.postgresql. org/download/postgresql-42.2.19.jar

Datenbank-Engine	Treiber	Speicherort für den Download
Amazon EMR	`HiveJDBC42.jar`	http://awssupportdatasvcs. com/bootstrap-actions/Simba/latest/
Amazon Redshift	`redshift-jdbc42-2.1.0.9.jar`	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift Serverless	`redshift-jdbc42-2.1.0.9.jar`	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Apache Hive	`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
Azure SQL Database	`mssql-jdbc-7.2.2.jre11.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/release- -Treiber? notes-for-the-jdbc ansicht= 15 #72 sql-server-ver
Azure Synapse-Analytik	`mssql-jdbc-7.2.2.jre11.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/releasenotes-for-the-jdbc- -Treiber? ansicht= 15 #72 sql-server-ver
Greenplum-Datenbank	`postgresql-42.2.19.jar`	https://jdbc.postgresql. org/download/postgresql-42.2.19.jar
IBM Db2 für z/OS	`db2jcc-db2jcc4.jar`	https://www.ibm.com/support/Seiten/db2- -downloads-db2-zos jdbc-driver-versions-and
IBM Db2 (LUW)	`db2jcc-db2jcc4.jar`	https://www.ibm.com/support/Seiten/Node/382667
MariaDB	`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

Datenbank-Engine	Treiber	Speicherort für den Download
Microsoft SQL Server	`mssql-jdbc-10.2.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/download- -SQL-Server? microsoft-jdbc-driver-for sql-server-verAnsicht= 15 **Note** AWS SCT unterstützt nicht die neueste JDBC-Treiberversion 18.2.1.0 für MSSQL. Wir empfehlen die Installation des JDBC-Treibers Version 10.2 `mssql-jdbc-10.2.jar`).
MySQL	`mysql-connector-java-8.0.15.jar`	https://dev.mysql.com/downloads/Konnektor/j/
Netezza	`nzjdbc.jar` Verwenden Sie die Client-Tools-Software. Laden Sie die Treiberversion 7.2.1 herunter, die abwärtskompatibel mit Data Warehouse-Version 7.2.0 ist.	http://www.ibm. com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html
Oracle	`ojdbc8.jar` Treiberversionen 8 und höher werden unterstützt.	https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html
PostgreSQL	`postgresql-42.2.19.jar`	https://jdbc.postgresql. org/download/postgresql-42.2.19.jar
SAP ASE (Sybase ASE)	`jconn4.jar`	Der jConnect JDBC-Treiber

Datenbank-Engine	Treiber	Speicherort für den Download
Snowflake	`snowflake-jdbc-3.9.2.jar` Weitere Informationen finden Sie unter Den JDBC-Treiber herunterladen/integrieren.	https://repo1.maven. org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar

Teradata	`terajdbc4.jar`
	
	`tdgssconfig.jar`
	
	Für Teradata JDBC-Treiberversion 16.20.00.11 und höher benötigen Sie die Datei `tdgssconfig.jar` nicht.
		https://downloads.teradata.com/download/Konnektivität/JDBC-Treiber

Vertica	`vertica-jdbc-9.1.1-0.jar` Treiberversionen 7.2.0 und höher werden unterstützt.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Installation von JDBC-Treibern unter Linux

Sie können die folgenden Schritte ausführen, um die JDBC-Treiber auf Ihrem Linux-System zur Verwendung mit zu installieren. AWS SCT

So installieren Sie die JDBC-Treiber auf Ihrem Linux-System

1. Erstellen Sie ein Verzeichnis, in dem Sie die JDBC-Treiber speichern möchten.

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

2. Installieren Sie die JDBC-Treiber für Ihre Datenbank-Engine mit den folgenden Befehlen.

Datenbank-Engine	Befehle zur Installation
Amazon Aurora (kompatibel mit MySQL)	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz`
Amazon Aurora (kompatibel mit PostgreSQL)	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .`
Microsoft SQL Server	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz`
MySQL	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz`

Oracle	`PROMPT> cd /usr/local/jdbc-drivers`
	`PROMPT> sudo mkdir oracle-jdbc`
	`PROMPT> cd oracle-jdbc`
	`PROMPT> sudo cp  -a /tmp/ojdbc8.jar .`

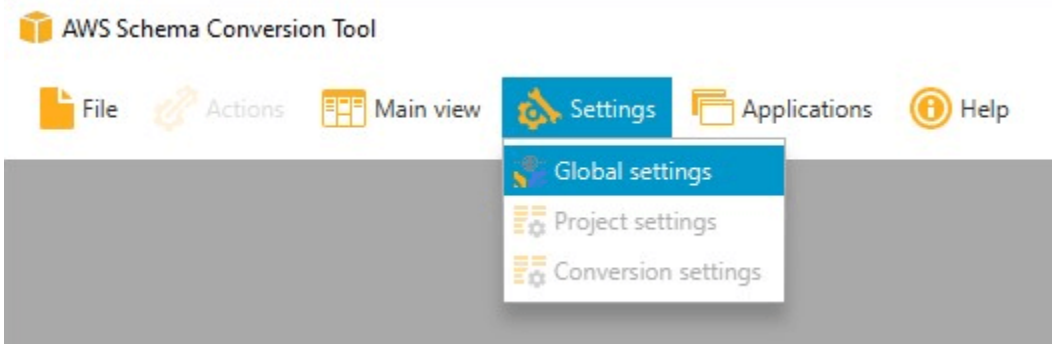
PostgreSQL	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .`

Speichern von Treiberpfaden in den globalen Einstellungen

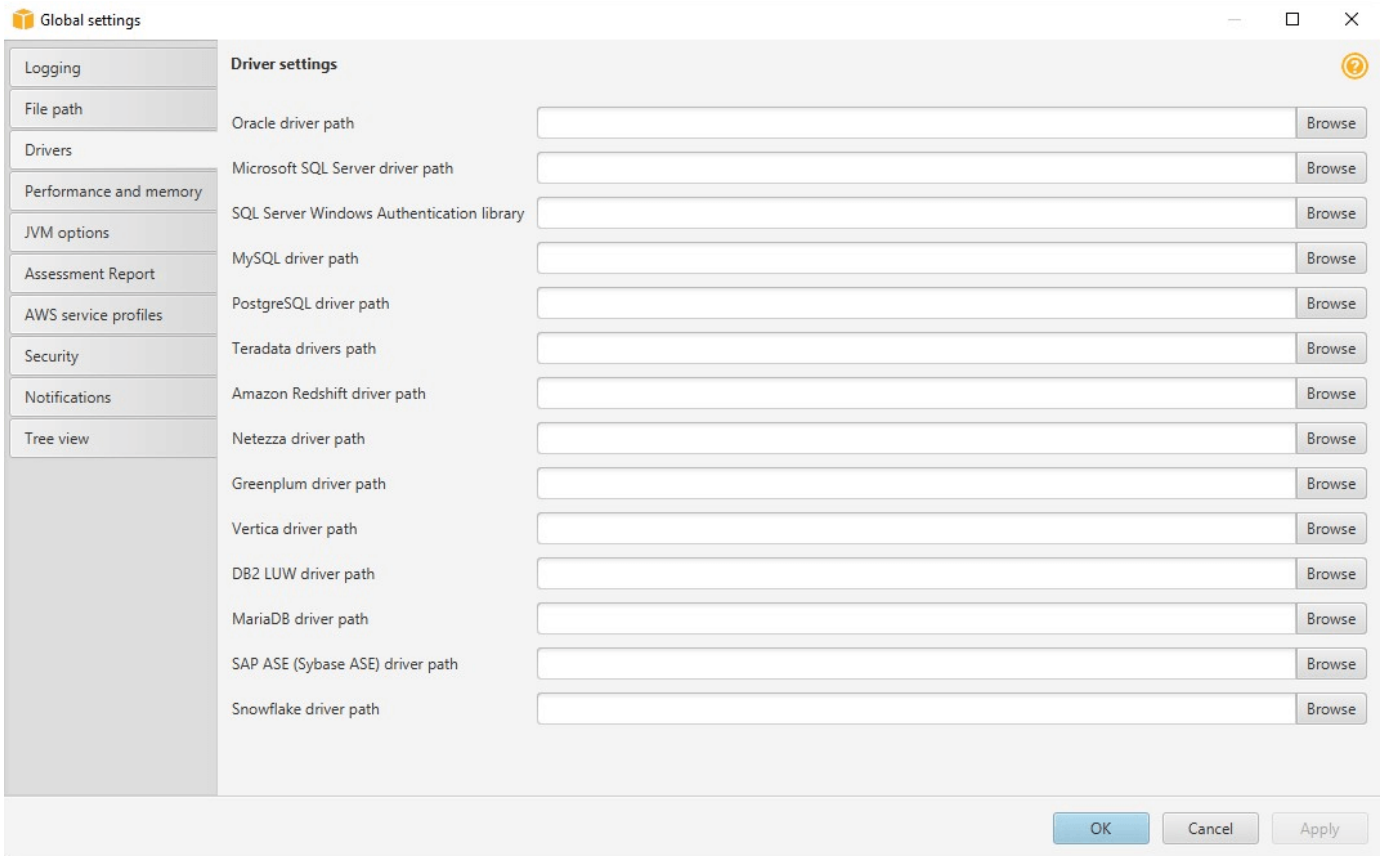
Nachdem Sie die erforderlichen JDBC-Treiber heruntergeladen und installiert haben, können Sie den Speicherort der Treiber global in den AWS SCT Einstellungen festlegen. Wenn Sie den Speicherort der Treiber nicht global festlegen, müssen Sie diesen bei jedem Verbindungsvorgang zu einer Datenbank in der Anwendung angeben.

So aktualisieren Sie den Speicherort der Treiberdatei

1. Wählen Sie in AWS SCTEinstellungen und anschließend Allgemeine Einstellungen aus.



2. Klicken Sie unter Global Settings auf Drivers. Fügen Sie den Dateipfad zum JDBC-Treiber für Ihre Quelldatenbank-Engine und Ihre Amazon RDS-DB-Instance-Ziel-Datenbank-Engine hinzu.



3. Wenn Sie die Treiberpfade hinzugefügt haben, klicken Sie auf OK.

Aktualisierung AWS Schema Conversion Tool

AWS wird regelmäßig AWS SCT mit neuen Features und Funktionen aktualisiert. Wenn Sie von einer früheren Version aktualisieren, erstellen Sie ein neues AWS SCT Projekt und konvertieren Sie alle Datenbankobjekte, die Sie verwenden, erneut.

Sie können überprüfen, ob Updates für AWS SCT vorhanden sind.

Um nach Updates zu suchen AWS SCT

1. Wenn Sie angemeldet sind AWS SCT, wählen Sie Hilfe und dann Nach Updates suchen.
2. Wählen Sie im Dialogfeld Check for Updates (Nach Updates suchen) die Option What's New (Neues) aus. Wenn der Link nicht angezeigt wird, haben Sie die neueste Version.

AWS Schema Conversion Tool CLI

Sie können die AWS SCT CLI für die Befehlszeilenverwendung herunterladen. Verwenden Sie den folgenden Link, um das JAR herunterzuladen:

AWSSchemaConversionToolBatch.jar

Navigieren in der Benutzeroberfläche des AWS SCT

Die folgenden Themen sollen Ihnen bei der Arbeit mit der AWS SCT Benutzeroberfläche helfen. Informationen zur Installation finden AWS SCT Sie unterInstallation und Konfiguration AWS Schema Conversion Tool.

Themen

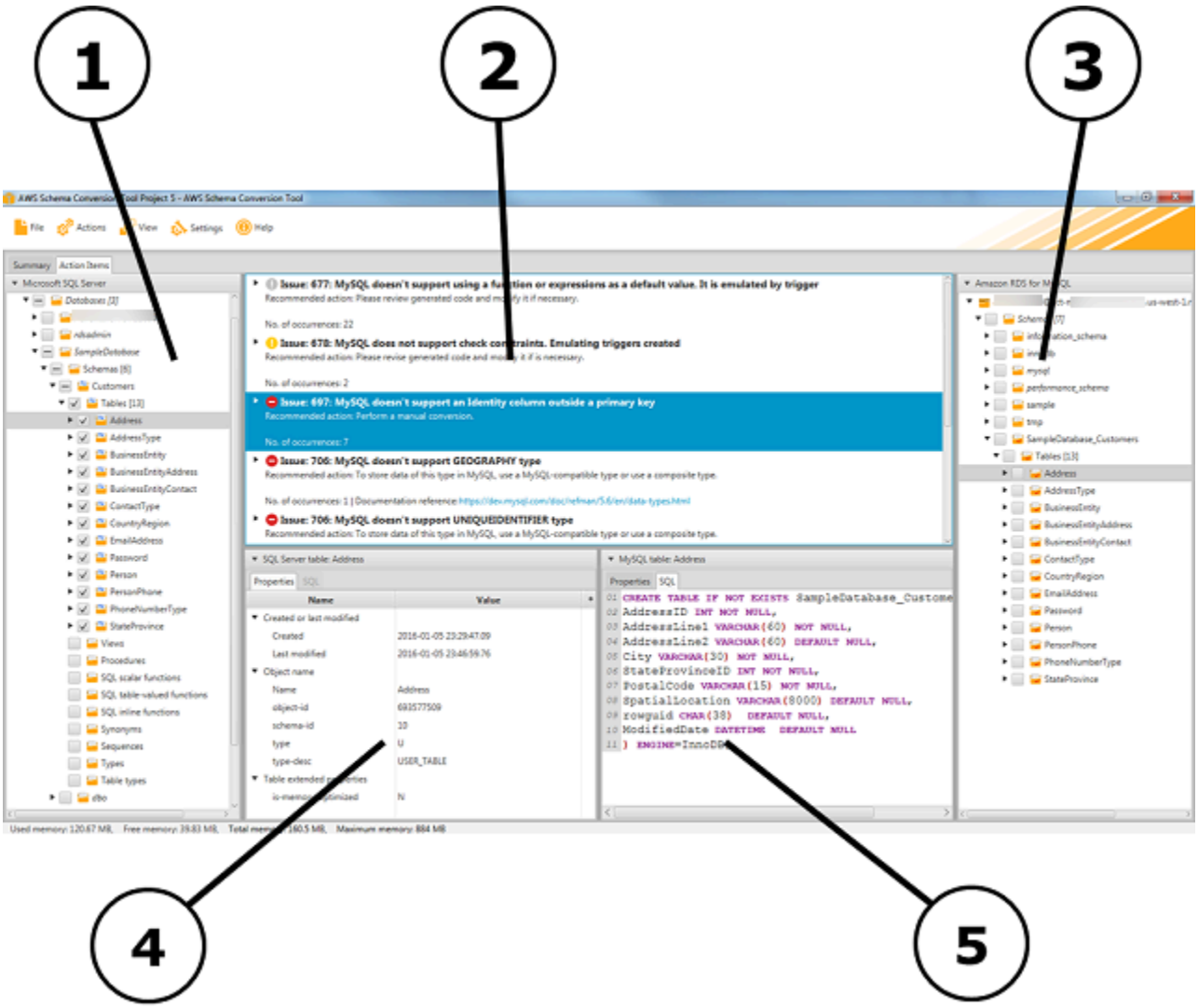
- Das Projektfenster anzeigen in AWS SCT
- Projekte starten und verwalten in AWS SCT
- Verwenden Sie den Assistenten AWS SCT
- Projekte speichern in AWS SCT
- Server zum Projekt hinzufügen in AWS SCT
- Verwenden Sie den Offline-Modus in AWS Schema Conversion Tool
- Verwenden von Baumfiltern in AWS Schema Conversion Tool
- Schemas verstecken in AWS Schema Conversion Tool
- Den Bewertungsbericht anzeigen in AWS Schema Conversion Tool
- Schemas konvertieren in AWS Schema Conversion Tool
- Anwenden der konvertierten Schemas in AWS Schema Conversion Tool
- Profile verwalten in AWS Schema Conversion Tool
- Konfiguration AWS Secrets Manager in der AWS Schema Conversion Tool
- Speichern von Passwörtern im AWS Schema Conversion Tool
- Die UNION ALL-Ansicht wird erstellt in AWS Schema Conversion Tool
- Verwenden von Tastenkombinationen im AWS Schema Conversion Tool

Das Projektfenster anzeigen in AWS SCT

Die folgende Abbildung zeigt, was Sie sehen, AWS SCT wenn Sie ein Schemamigrationsprojekt erstellen und anschließend ein Schema konvertieren.

1. Im linken Bereich wird das Schema der Quelldatenbank in einer Strukturansicht präsentiert. Ihr Datenbankschema ist "lazy loaded". Mit anderen Worten, wenn Sie ein Element aus der Strukturansicht auswählen, AWS SCT wird das aktuelle Schema aus Ihrer Quelldatenbank abgerufen und angezeigt.

2. Im oberen mittleren Bereich werden Aktionselemente für Schemaelemente aus der Quelldatenbank-Engine angezeigt, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden konnten.
3. Im rechten Bereich wird das Schema der Ziel-DB-Instance in einer Strukturansicht präsentiert. Ihr Datenbankschema ist "lazy loaded". Das heißt, wenn Sie ein Element aus der Strukturansicht auswählen, AWS SCT wird das aktuelle Schema aus Ihrer Zieldatenbank abgerufen und angezeigt.



4. Wenn Sie im unteren linken Bereich ein Schemaelement auswählen, werden Eigenschaften angezeigt. Diese beschreiben das Quellschemaelement und den SQL-Befehl zum Erstellen dieses Elements in der Quelldatenbank.

5. Wenn Sie im unteren rechten Bereich ein Schemaelement auswählen, werden Eigenschaften angezeigt. Diese beschreiben das Zielschemaelement und den SQL-Befehl zum Erstellen dieses Elements in der Zieldatenbank. Sie können diesen SQL-Befehl bearbeiten und die aktualisierte Version in Ihrem Projekt speichern.

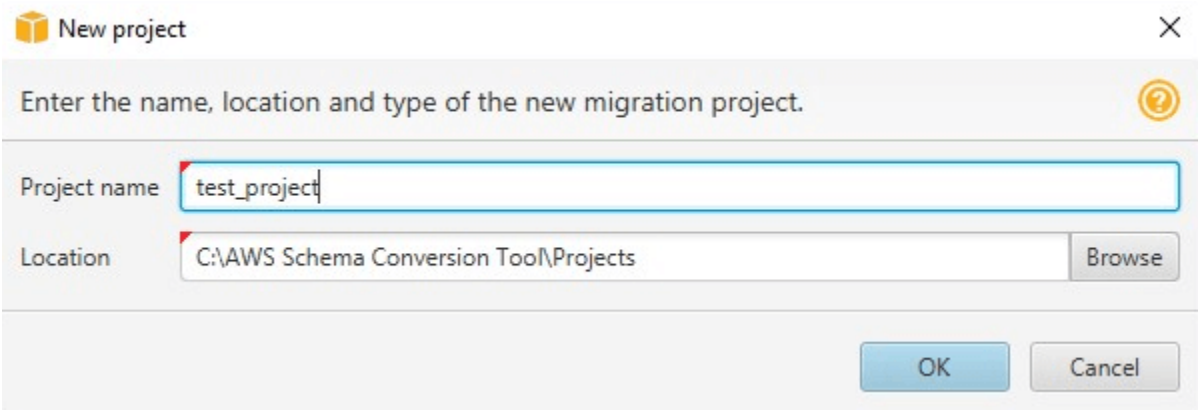
Projekte starten und verwalten in AWS SCT

Doppelklicken Sie auf das Anwendungssymbol AWS Schema Conversion Tool, um das zu starten.

Gehen Sie wie folgt vor, um ein AWS Schema Conversion Tool Projekt zu erstellen.

So erstellen Sie Ihr Projekt

1. Starten Sie das AWS Schema Conversion Tool.
2. Wählen Sie im Menü Datei die Option Neues Projekt aus. Das Dialogfeld „Neues Projekt“ wird angezeigt.



3. Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.
4. Geben Sie den Speicherort für Ihre lokale Projektdatei an.
5. Wählen Sie OK, um Ihr AWS SCT Projekt zu erstellen.
6. Wählen Sie Quelle hinzufügen, um Ihrem AWS SCT Projekt eine neue Quelldatenbank hinzuzufügen. Sie können Ihrem AWS SCT Projekt mehrere Quelldatenbanken hinzufügen.
7. Wählen Sie Ziel hinzufügen, um Ihrem AWS SCT Projekt eine neue Zielplattform hinzuzufügen. Sie können Ihrem AWS SCT Projekt mehrere Zielplattformen hinzufügen.
8. Wählen Sie das Quelldatenbankschema im linken Bereich aus.
9. Geben Sie im rechten Bereich die Zieldatenbankplattform für das ausgewählte Quellschema an.

10. Wählen Sie Zuordnung erstellen. Diese Schaltfläche wird aktiv, nachdem Sie das Quelldatenbankschema und die Zieldatenbankplattform ausgewählt haben. Weitere Informationen finden Sie unter Zuordnung von Datentypen.

Jetzt ist Ihr AWS SCT Projekt eingerichtet. Sie können Ihr Projekt speichern, einen Bewertungsbericht zur Datenbankmigration erstellen und Ihre Quelldatenbankschemas konvertieren.

Verwenden Sie den Assistenten AWS SCT

Mit dem Assistenten für neue Projekte können Sie ein neues Datenbankmigrationsprojekt erstellen. Dieser Assistent unterstützt Sie dabei, Ihr Migrationsziel zu bestimmen und eine Verbindung zu Ihren Datenbanken herzustellen. Er schätzt, wie komplex eine Migration für alle unterstützten Zielziele sein könnte. Nachdem Sie den Assistenten ausgeführt haben, wird ein zusammenfassender Bericht über die Migration Ihrer Datenbank zu verschiedenen Zielorten AWS SCT erstellt. Sie können diesen Bericht verwenden, um mögliche Zielziele zu vergleichen und den optimalen Migrationspfad auszuwählen.

Um den Assistenten für neue Projekte auszuführen

1. Wählen Sie Ihre Quelldatenbank aus.

   a. Starten Sie das AWS Schema Conversion Tool.

   b. Wählen Sie im Menü Datei die Option Assistent für neues Projekt aus. Das Dialogfeld Neues Datenbankmigrationsprojekt erstellen wird geöffnet.

   c. Gehen Sie wie folgt vor, um die Verbindungsinformationen für die Quelldatenbank einzugeben:

Parameter	Aktion
Project name	Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.
Ort	Geben Sie den Speicherort für Ihre lokale Projektdatei an.
Source type (Quellentyp)	Wählen Sie eine der folgenden Optionen: SQL-Datenbank, NoSQL-Datenbank oder ETL.

Parameter	Aktion
	Wenn Sie den zusammenfassenden Bericht mit allen Migration szielen sehen möchten, wählen Sie SQL-Datenbank.
Source engine (Quell-Engine)	Wählen Sie Ihre Quelldatenbank-Engine aus.
Migrationsstrategie	Wählen Sie eine der folgenden Optionen: • Ich möchte zwischen den Engines wechseln und für die Cloud optimieren — Diese Option konvertiert Ihre Quelldatenbank in eine neue Datenbank-Engine. • Ich möchte dieselbe Engine beibehalten, aber für die Cloud optimieren — Bei dieser Option bleibt Ihre Datenbank-Engine unverändert und die Datenbank wird von der lokalen Datenbank in die Cloud verschoben. • Ich möchte einen kombinierten Bericht für den Wechsel der Datenbank-Engine und die Optimierung für die Cloud sehen. Bei dieser Option wird die Migrationskomplexität aller verfügbaren Migrationsoptionen verglichen. Wenn Sie den aggregierten Bewertungsbericht sehen möchten, der alle Migrationsziele umfasst, wählen Sie die letzte Option.

d. Wählen Sie Weiter. Die Seite Mit der Quelldatenbank Connect wird geöffnet.

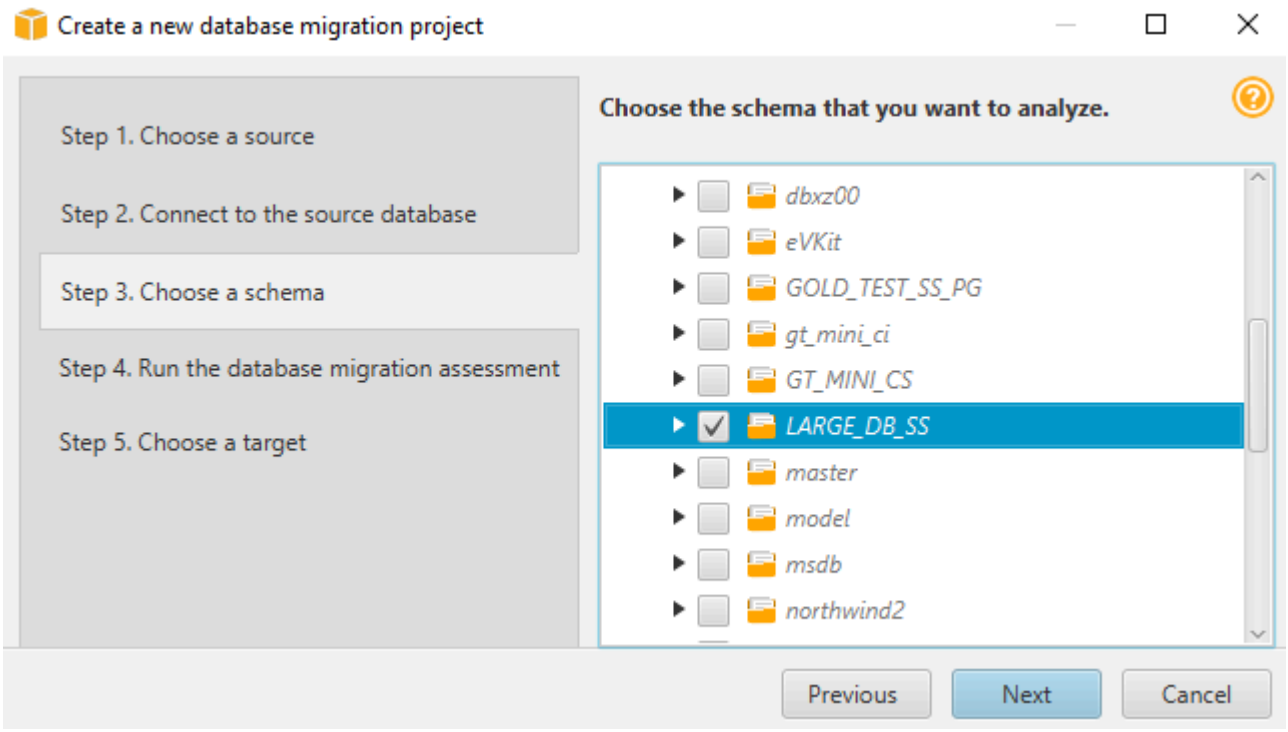
2. Stellen Sie eine Verbindung zur Quelldatenbank her.

   a. Geben Sie Ihre Verbindungsinformationen für die Quelldatenbank an. Die Verbindungsparameter hängen von Ihrer Quelldatenbank-Engine ab. Stellen Sie sicher, dass der Benutzer, den Sie für die Analyse Ihrer Quelldatenbank verwenden, über die entsprechenden Berechtigungen verfügt. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool.

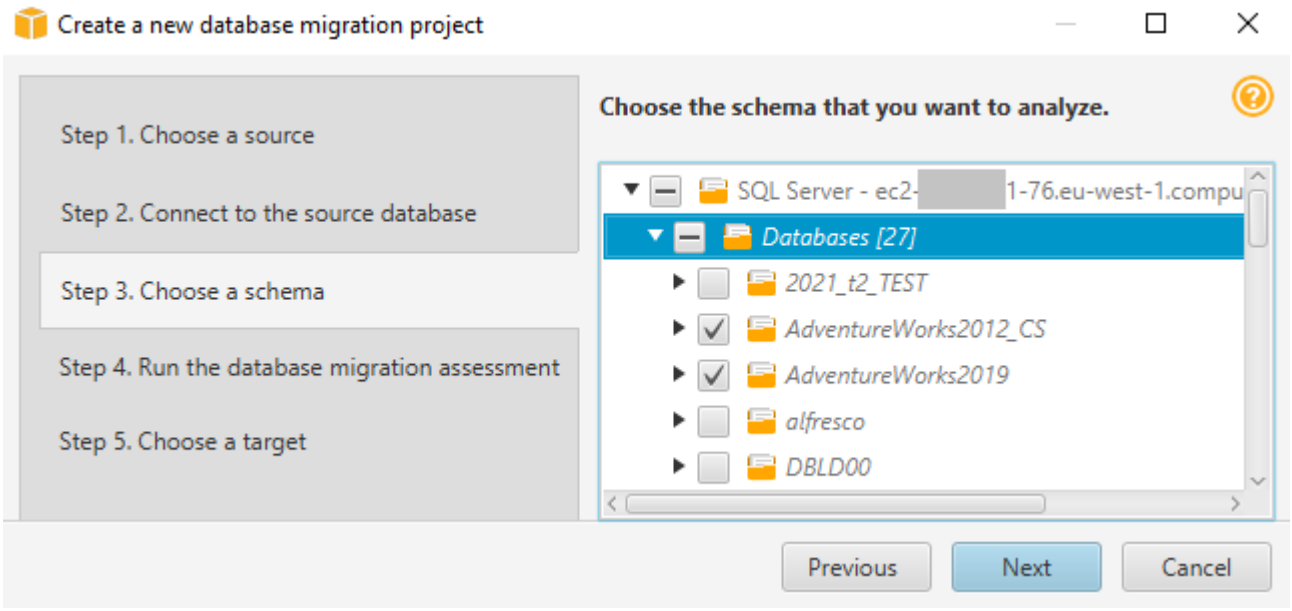
   b. Wählen Sie Weiter. Die Seite „Schema auswählen“ wird geöffnet.

3. Wählen Sie Ihr Datenbankschema aus.

a. Aktivieren Sie das Kontrollkästchen für die Namen der Schemas, die Sie bewerten möchten, und wählen Sie dann das Schema selbst aus. Der Schemaname wird blau hervorgehoben, wenn er ausgewählt ist und die Schaltfläche Weiter verfügbar ist.



b. Wenn Sie mehrere Datenbankschemas bewerten möchten, aktivieren Sie die Kontrollkästchen für alle Schemas und wählen Sie dann den übergeordneten Knoten aus. Für eine erfolgreiche Bewertung müssen Sie den übergeordneten Knoten auswählen. Wählen Sie beispielsweise für eine SQL Server-Quelldatenbank den Knoten Datenbanken aus. Der Name des übergeordneten Knotens ist blau hervorgehoben und die Schaltfläche Weiter ist verfügbar.



c. Wählen Sie Weiter. AWS SCT analysiert Ihre Quelldatenbankschemas und erstellt einen Bewertungsbericht zur Datenbankmigration. Die Anzahl der Datenbankobjekte in Ihren Quelldatenbankschemas wirkt sich auf die Zeit aus, die für die Durchführung der Bewertung benötigt wird. Wenn der Vorgang abgeschlossen ist, wird die Seite Bewertung der Datenbankmigration ausführen geöffnet.

4. Führen Sie die Bewertung der Datenbankmigration aus.

   a. Sie können die Bewertungsberichte für verschiedene Migrationsziele überprüfen und vergleichen oder eine lokale Kopie der Bewertungsberichtsdateien für die weitere Analyse speichern.

   b. Speichern Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration. Wählen Sie Speichern, geben Sie dann den Pfad zu dem Ordner ein, in dem die Dateien gespeichert werden sollen, und wählen Sie Speichern. AWS SCT speichert die Bewertungsberichtsdateien im angegebenen Ordner.

   c. Wählen Sie Weiter. Die Seite „Ziel auswählen“ wird geöffnet.

5. Wählen Sie Ihre Zieldatenbank aus.

   a. Wählen Sie für Target Engine die Zieldatenbank-Engine aus, die Sie auf der Grundlage des Bewertungsberichts verwenden möchten.

   b. Geben Sie Ihre Verbindungsinformationen für Ihre Zieldatenbank ein. Die angezeigten Verbindungsparameter hängen von der ausgewählten Zieldatenbank-Engine ab. Stellen Sie sicher, dass der für die Zieldatenbank angegebene Benutzer über die erforderlichen Berechtigungen verfügt. Weitere Informationen zu den erforderlichen Berechtigungen finden

Sie in den Abschnitten, in denen die Berechtigungen für Zieldatenbanken beschrieben werden, unter Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool undBerechtigungen für Amazon Redshift als Ziel.

c. Wählen Sie Fertig stellen. AWS SCT erstellt Ihr Projekt und fügt die Zuordnungsregeln hinzu. Weitere Informationen finden Sie unter Zuordnung von Datentypen.

Jetzt können Sie das AWS SCT Projekt verwenden, um Ihre Quelldatenbankobjekte zu konvertieren.

Projekte speichern in AWS SCT

Gehen Sie wie folgt vor, um ein AWS Schema Conversion Tool Projekt zu speichern.

Um Ihr Projekt zu speichern

1. Starte das AWS Schema Conversion Tool.
2. Wählen Sie im Menü Datei die Option Projekt speichern.

   AWS SCT speichert das Projekt in dem Ordner, den Sie bei der Erstellung des Projekts angegeben haben.

Gehen Sie wie folgt vor, um ein vorhandenes AWS Schema Conversion Tool Projekt zu öffnen.

Um Ihr Projekt zu öffnen

1. Wählen Sie im Menü Datei die Option Projekt öffnen. Das Dialogfeld Öffnen wird angezeigt.
2. Wählen Sie den Projektordner und dann die Windows Script Component-Datei (*.sct) aus.
3. AWS SCT öffnet Ihr Projekt, stellt jedoch nicht automatisch eine Verbindung zu Ihren Quell- und Zieldatenbanken her. Wählen Sie oben in Ihren Datenbank-Schemabäumen die Option Mit dem Server verbinden, um eine Verbindung zu Ihren Quell- und Zieldatenbanken herzustellen.

Wenn Sie ein Projekt öffnen, das in AWS SCT Version 1.0.655 oder früher gespeichert wurde, erstellt es AWS SCT automatisch Zuordnungsregeln für alle Quelldatenbankschemas zur Zieldatenbankplattform. Um weitere Zieldatenbankplattformen hinzuzufügen, löschen Sie bestehende Zuordnungsregeln und erstellen Sie dann neue Zuordnungsregeln. Weitere Informationen zum Erstellen von Zuordnungsregeln finden Sie unterZuordnung von Datentypen.

Server zum Projekt hinzufügen in AWS SCT

Sie können einem AWS Schema Conversion Tool Projekt mehrere Quell- und Zieldatenbankserver hinzufügen.

Um Ihrem Projekt einen Server hinzuzufügen

1. Starte das AWS Schema Conversion Tool.
2. Erstellen Sie ein neues Projekt oder öffnen Sie ein vorhandenes Projekt.
3. Wählen Sie Quelle hinzufügen aus dem Menü, um eine neue Quelldatenbank hinzuzufügen.
4. Wählen Sie eine Datenbankplattform und geben Sie die Anmeldeinformationen für die Datenbankverbindung an. Weitere Hinweise zum Herstellen einer Verbindung mit einer Quelldatenbank finden Sie unterVerbindung zu Quelldatenbanken herstellen.

Verwenden Sie das folgende Verfahren, um eine Verbindung zu Ihrer Datenbank herzustellen.

Um eine Verbindung zu Ihrer Datenbank herzustellen

1. Öffnen Sie das Kontextmenü (Rechtsklick) für einen Datenbankserver und wählen Sie dann Verbindung herstellen.

   Sie können auch oben in Ihrer Datenbankschemabaumstruktur die Option Mit dem Server Connect auswählen.
2. Geben Sie das Passwort ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.
3. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
4. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Gehen Sie wie folgt vor, um einen Datenbankserver aus Ihrem AWS SCT Projekt zu entfernen.

Um einen Datenbankserver zu entfernen

1. Wählen Sie den Datenbankserver aus, den Sie entfernen möchten.
2. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Aus Projekt entfernen.

   AWS SCT entfernt den ausgewählten Datenbankserver, alle Zuordnungsregeln, Konvertierungsergebnisse und andere Metadaten, die sich auf diesen Server beziehen.

Verwenden Sie den Offline-Modus in AWS Schema Conversion Tool

Sie können es AWS Schema Conversion Tool im Offline-Modus ausführen. Im Folgenden erfahren Sie, wie Sie mit einem vorhandenen AWS SCT Projekt arbeiten, wenn die Verbindung zu Ihrer Quelldatenbank getrennt ist.

AWS SCT benötigt keine Verbindung zu Ihrer Quelldatenbank, um die folgenden Operationen auszuführen:

- Fügen Sie Zuordnungsregeln hinzu.
- Erstellen Sie Bewertungsberichte zur Datenbankmigration.
- Konvertieren Sie Datenbankschemas und Code.
- Bearbeiten Sie Ihren Quellcode und den konvertierten Code.
- Speichern Sie Ihren Quellcode und den konvertierten Code als SQL-Skripts in einer Textdatei.

Stellen Sie vor AWS SCT der Verwendung im Offline-Modus eine Verbindung zu Ihrer Quelldatenbank her, laden Sie Metadaten und speichern Sie Ihr Projekt. Öffnen Sie dieses Projekt oder trennen Sie die Verbindung zum Quelldatenbankserver, um es AWS SCT im Offlinemodus zu verwenden.

Um es AWS SCT im Offline-Modus auszuführen

1. Starten Sie das AWS Schema Conversion Tool und erstellen Sie ein neues Projekt. Weitere Informationen finden Sie unter Projekte starten und verwalten in AWS SCT.
2. Fügen Sie einen Quelldatenbankserver hinzu und stellen Sie eine Verbindung zu Ihrer Quelldatenbank her. Weitere Informationen finden Sie unter Server zum Projekt hinzufügen in AWS SCT.
3. Fügen Sie einen Zieldatenbankserver hinzu oder verwenden Sie eine virtuelle Zieldatenbankplattform. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
4. Erstellen Sie eine Zuordnungsregel, um die Zieldatenbankplattform für Ihre Quelldatenbank zu definieren. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.
5. Wählen Sie Ansicht und anschließend Hauptansicht.

6. Wählen Sie im linken Bereich, in dem die Objekte Ihrer Quelldatenbank angezeigt werden, Ihre Quelldatenbankschemas aus. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie dann Schema laden. Dieser Vorgang lädt alle Metadaten des Quellschemas in Ihr AWS SCT Projekt.

   Mit den Vorgängen Bericht erstellen und Schema konvertieren werden auch alle Metadaten des Quellschemas in Ihr AWS SCT Projekt geladen. Wenn Sie einen dieser Vorgänge über das Kontextmenü ausgeführt haben, überspringen Sie den Vorgang Schema laden.
7. Wählen Sie im Menü Datei die Option Projekt speichern, um die Metadaten der Quelldatenbank in Ihrem Projekt zu speichern.
8. Wählen Sie Verbindung zum Server trennen, um die Verbindung zu Ihrer Quelldatenbank zu trennen. Jetzt können Sie es AWS SCT im Offline-Modus verwenden.

Verwenden von Baumfiltern in AWS Schema Conversion Tool

Um Daten von einer Quelle zu einem Ziel zu migrieren, werden alle Metadaten aus Quell- und Zieldatenbanken in eine Baumstruktur AWS SCT geladen. Diese Struktur wird in AWS SCT als Baumstrukturansicht im Hauptfenster angezeigt.

Einige Datenbanken können sehr viele Objekte in der Baumstruktur haben. Sie können Baumfilter verwenden AWS SCT , um nach Objekten in den Quell- und Zielbaumstrukturen zu suchen. Wenn Sie einen Baumstrukturfilter verwenden, ändern Sie nicht die Objekte, die konvertiert werden, wenn Sie Ihre Datenbank konvertieren. Der Filter ändert nur das, was Sie in der Baumstruktur sehen.

Baumfilter funktionieren mit Objekten, die vorinstalliert AWS SCT wurden. Mit anderen Worten: AWS SCT lädt bei der Suche keine Objekte aus der Datenbank. Dieser Ansatz bedeutet, dass die Baumstruktur im Allgemeinen weniger Objekte enthält, als in der Datenbank vorhanden sind.

Beachten Sie für Baumstrukturfilter Folgendes:

- Der Standardwert für den Filter ist ANY, was bedeutet, dass der Filter eine Namenssuche verwendet, um Objekte zu finden.
- Wenn Sie einen oder mehrere Objekttypen auswählen, sehen Sie nur diese Objekttypen in der Baumstruktur.
- Sie können den Filter verwenden, um unterschiedliche Symboltypen anzuzeigen, einschließlich Unicode, Leerzeichen und Sonderzeichen. "%" ist der Platzhalter für alle Zeichen.

- Nachdem Sie einen Filter angewendet haben, zeigt der Zähler nur die Anzahl der gefilterten Objekte.

So erstellen Sie einen Baumstrukturfilter

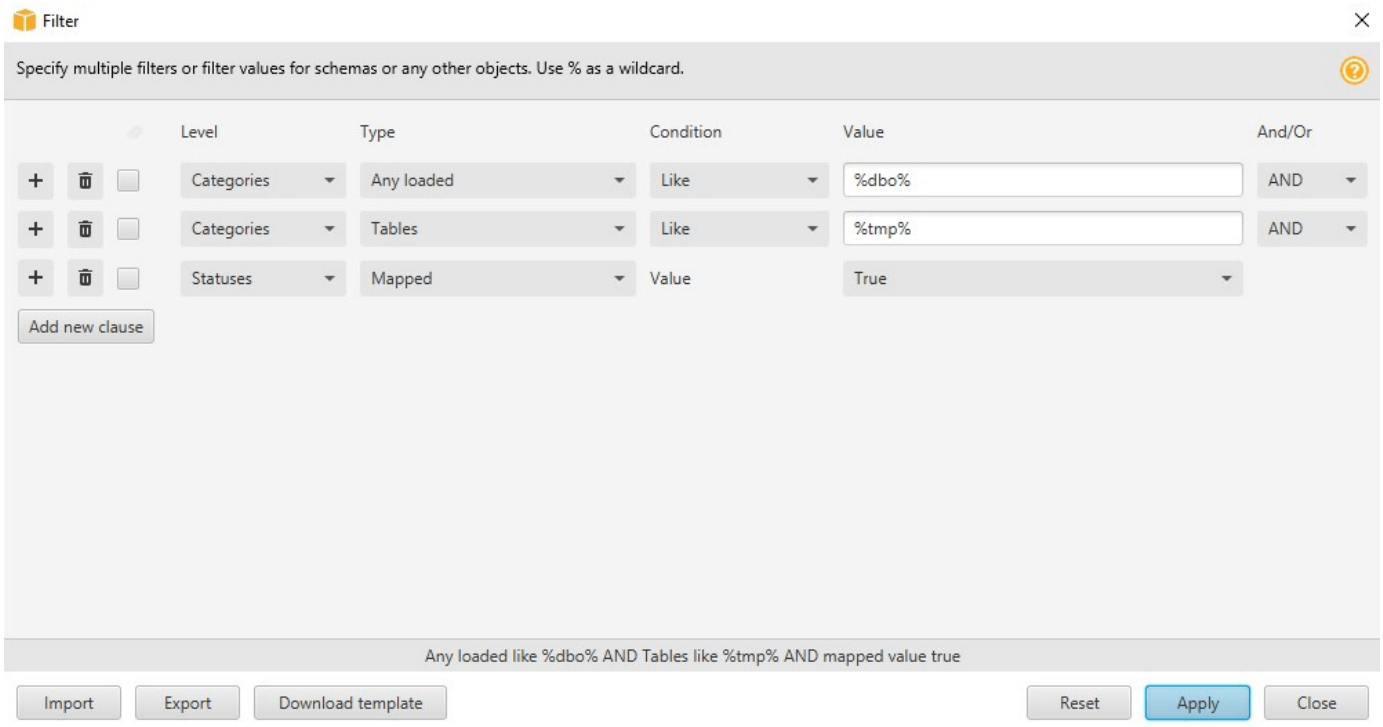
1. Öffnen Sie ein vorhandenes AWS SCT Projekt.
2. Connect zu der Datenbank her, auf die Sie den Baumfilter anwenden möchten.
3. Wählen Sie das Filtersymbol.

   Das Symbol für das Aufheben des Filters ist deaktiviert, da aktuell kein Filter angewendet wird.
4. Geben Sie die folgenden Informationen in das Dialogfeld Filter ein. Die Optionen im Dialogfeld unterscheiden je nach Datenbank-Engine.

AWS SCT Filteroption	Aktion
Stufe	Wählen Sie Kategorien, um Objekte nach Kategorien zu filtern. Wählen Sie Status, um Objekte nach Status zu filtern.
Typ	Wählen Sie unter Kategorien in Level die Kategorien der gefilterten Objekte aus. Wählen Sie Alle geladenen, um Objekte aus allen Kategorien anzuzeigen. Wählen Sie unter Status in Level den Status der gefilterten Objekte aus. Sie können eine der folgenden Optionen wählen: • Konvertiert, um alle konvertierten Objekte anzuzeigen • Hat Aktionen zum Anzeigen aller Objekte, bei denen Konvertierungsprobleme auftreten • Verschlüsselt, um alle verschlüsselten Objekte anzuzeigen
Bedingung	Wählen Sie für Kategorien in Level die Filterbedingung zwischen „Gefällt mir“ und „Gefällt mir nicht“.

AWS SCT Filteroption	Aktion
	Für Status in Level ist die Option „Filterbedingung“ nicht verfügbar.
Wert	Geben Sie für Kategorien in Level den Wert ein, um die Struktur nach diesem Wert zu filtern. Verwenden Sie den Prozentsatz (%) als Platzhalter, um alle Objekte anzuzeigen. Wählen Sie für Status in Level einen Wert zwischen Wahr und Falsch.
Und/oder	Wählen Sie AND oder OR logische Operatoren, um mehrere Filterkla useln anzuwenden.



5. Wählen Sie Neue Klausel hinzufügen, um eine zusätzliche Filterklausel hinzuzufügen. AWS SCT kann mehrere Filterklauseln mithilfe von AND OR logischen Operatoren anwenden.

6. Wählen Sie Anwenden aus. Nachdem Sie Apply ausgewählt haben, wird das Symbol für die Aufhebung des Filters (neben dem Filtersymbol) aktiviert. Verwenden Sie dieses Symbol, wenn Sie die angewendeten Filter entfernen möchten.
7. Wählen Sie Close (Schließen), um das Dialogfeld zu schließen.

Wenn Sie das Schema filtern, das in der Struktur angezeigt wird, ändern Sie nicht die Objekte, die konvertiert werden, wenn Sie das Schema konvertieren. Der Filter ändert nur das, was in der Struktur zu sehen ist.

Eine Dateiliste für den Baumfilter importieren

Sie können eine Datei mit kommagetrennten Werten (CSV) mit Semikolontrennzeichen oder eine JSON-Datei importieren, die Namen oder Werte enthält, die der Baumfilter verwenden soll. Öffnen Sie ein vorhandenes AWS SCT Projekt, stellen Sie eine Verbindung mit der Datenbank her, auf die der Baumfilter angewendet werden soll, und wählen Sie dann das Filtersymbol aus.

Um ein Beispiel für die Datei herunterzuladen, wählen Sie Vorlage herunterladen. Geben Sie den Dateinamen ein und wählen Sie Speichern.

Um Ihre vorhandenen Filtereinstellungen herunterzuladen, wählen Sie Exportieren. Geben Sie den Dateinamen ein und wählen Sie Speichern.

Um eine Dateiliste für den Baumfilter zu importieren, wählen Sie Import. Wählen Sie eine zu importierende Datei aus und wählen Sie dann Open. Wählen Sie Apply und dann Close.

CSV-Dateien verwenden Semikolon als Trennzeichen und haben das folgende Format:

- `object_type` ist der Objekttyp, nach dem Sie suchen möchten.
- `database_name` ist der Name der Datenbank, in der das Objekt vorhanden ist.
- `schema_name` ist der Name des Schemas, in dem das Objekt vorhanden ist.
- `object_name` ist der Name des Objekts.
- `import_type`gibt an, ob `include` oder `exclude` dieses Element aus dem Filter ausgewählt werden soll.

Verwenden Sie JSON-Dateien, um komplexe Filterfälle zu beschreiben, z. B. verschachtelte Regeln. JSON-Dateien haben das folgende Format:

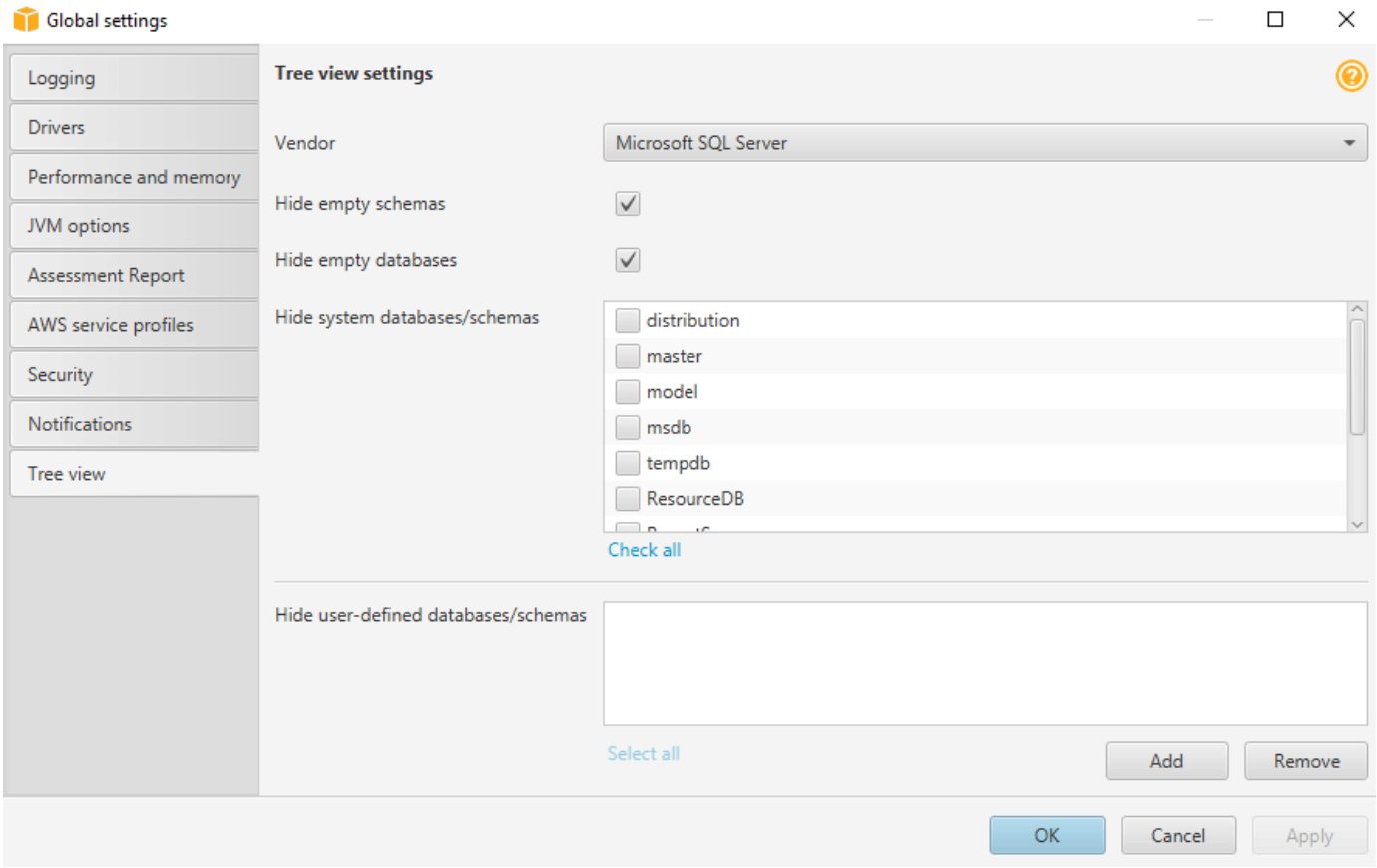
- `filterGroupType`ist der Typ der Filterregel (`AND`oder der `OR` logischen Operatoren), der für mehrere Filterklauseln gilt.
- `filterCategory`ist die Ebene des Filters (Kategorien oder Status).
- `names`ist die Liste der Objektnamen, die für den Filter Kategorien gilt.
- `filterCondition`ist die Filterbedingung (`LIKE`oder`NOT LIKE`), die für den Filter Kategorien gilt.
- `transformName`ist der Statusname, der für den Statusfilter gilt.
- `value`ist der Wert, nach dem der Baum gefiltert werden soll.
- `transformValue`ist der Wert des Filters (`TRUE`oder`FALSE`), der für den Statusfilter gilt.

Schemas verstecken in AWS Schema Conversion Tool

Verwenden Sie die Einstellungen für die Strukturansicht, um anzugeben, welche Schemas und Datenbanken Sie in der AWS SCT Baumansicht sehen möchten. Sie können leere Schemas, leere Datenbanken, System-Datenbanken und benutzerdefinierten Datenbanken und Schemas ausblenden.

So blenden Sie Datenbanken und Schemas in der Baumstrukturansicht aus

1. Öffnen Sie ein AWS SCT Projekt.
2. Stellen Sie eine Verbindung mit dem Datenspeicher her, den Sie in der Baumstrukturansicht anzeigen möchten.
3. Wählen Sie Einstellungen, Allgemeine Einstellungen, Baumansicht.



4. Gehen Sie im Bereich Einstellungen für die Strukturansicht wie folgt vor:

   - Wählen Sie als Anbieter die Datenbankplattform aus.
   - Wählen Sie Leere Schemas ausblenden, um leere Schemas für die ausgewählte Datenbankplattform auszublenden.
   - Wählen Sie Leere Datenbanken ausblenden, um leere Datenbanken für die ausgewählte Datenbankplattform auszublenden.
   - Wählen Sie unter Systemdatenbanken/Schemas ausblenden die Option Systemdatenbanken und Schemas nach Namen aus, um sie auszublenden.
   - Geben Sie unter Benutzerdefinierte Datenbanken/Schemas ausblenden die Namen der benutzerdefinierten Datenbanken und Schemas ein, die Sie ausblenden möchten, und wählen Sie dann Hinzufügen aus. Bei den Namen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

5. Wählen Sie OK aus.

Den Bewertungsbericht anzeigen in AWS Schema Conversion Tool

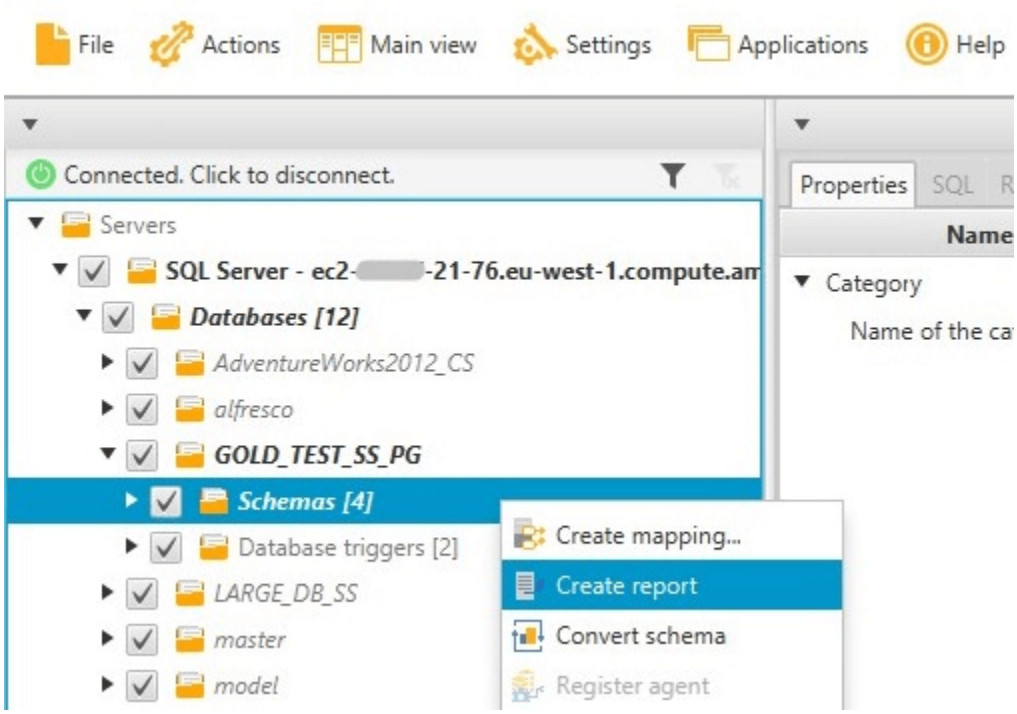
Der Bewertungsbericht zur Datenbankmigration fasst alle Aktionspunkte für Schemas zusammen, die nicht automatisch in die Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Im Bericht ist zudem eine Aufwandsschätzung für das Schreiben des äquivalenten Codes für Ihre Ziel-DB-Instance enthalten.

Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, nachdem Sie Ihrem Projekt die Quelldatenbanken und Zielplattformen hinzugefügt und die Zuordnungsregeln festgelegt haben.

So erstellen Sie den Bewertungsbericht zur Datenbankmigration und zeigen diesen an

1. Stellen Sie sicher, dass Sie eine Zuordnungsregel für das Quelldatenbankschema erstellt haben, für das Sie einen Bewertungsbericht erstellen möchten. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
2. Wählen Sie im Menü Ansicht die Option Hauptansicht.
3. Wählen Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, Schemaobjekte aus, für die Sie einen Bewertungsbericht erstellen möchten.

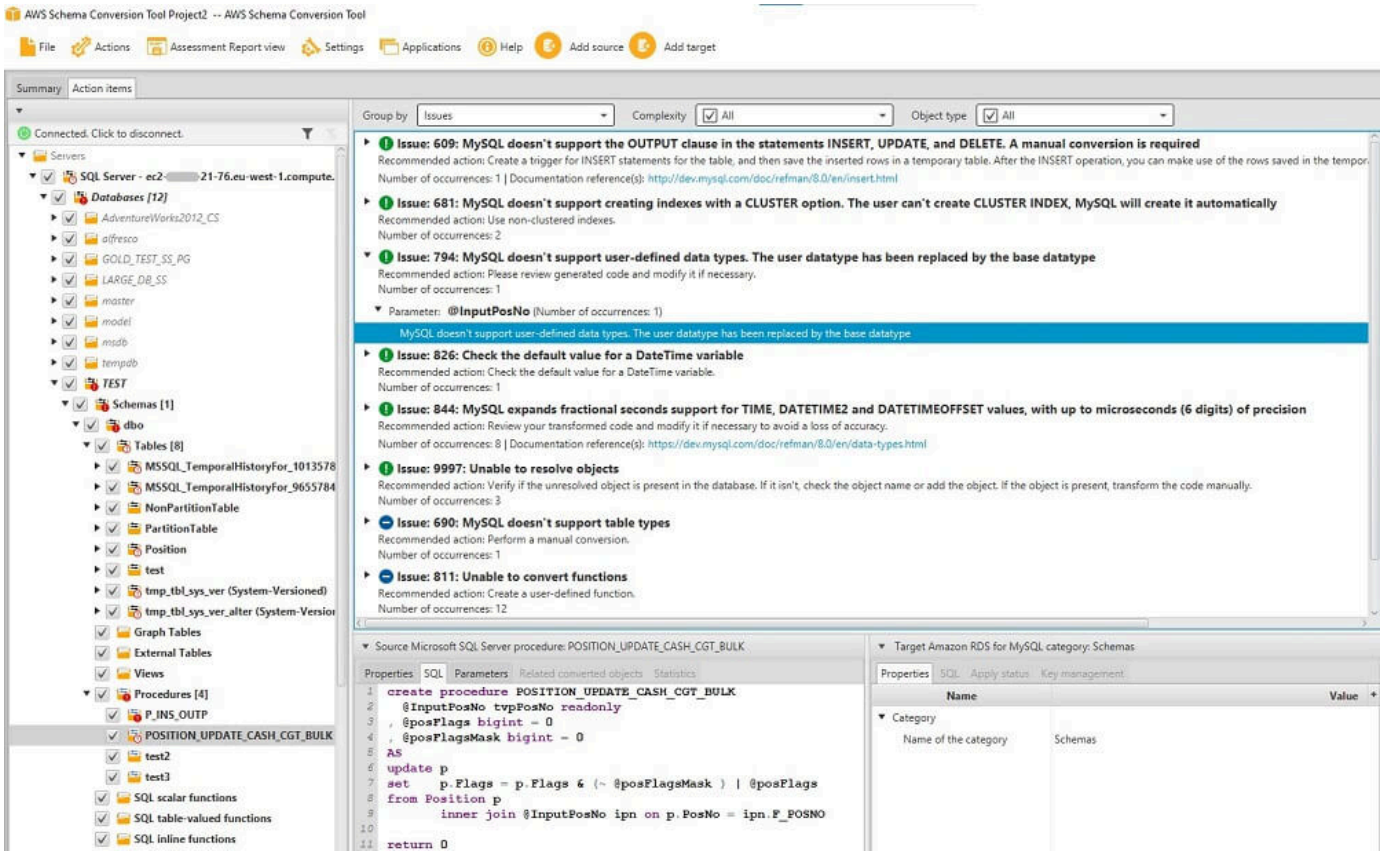
   Vergewissern Sie sich, dass Sie die Kontrollkästchen für alle Schemaobjekte aktiviert haben, für die Sie einen Bewertungsbericht erstellen möchten.
4. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie dann Bericht erstellen aus.



Die Bewertungsberichtansicht wird geöffnet.

5. Wählen Sie die Registerkarte Aktionselemente.

   Auf der Registerkarte Aktionspunkte wird eine Liste von Elementen angezeigt, die das Schema beschreiben, das nicht automatisch konvertiert werden kann. Wählen Sie einen der Aktionspunkte in der Liste aus. AWS SCT hebt das Element aus Ihrem Schema hervor, für das sich der Aktionspunkt bezieht, wie im Folgenden dargestellt.

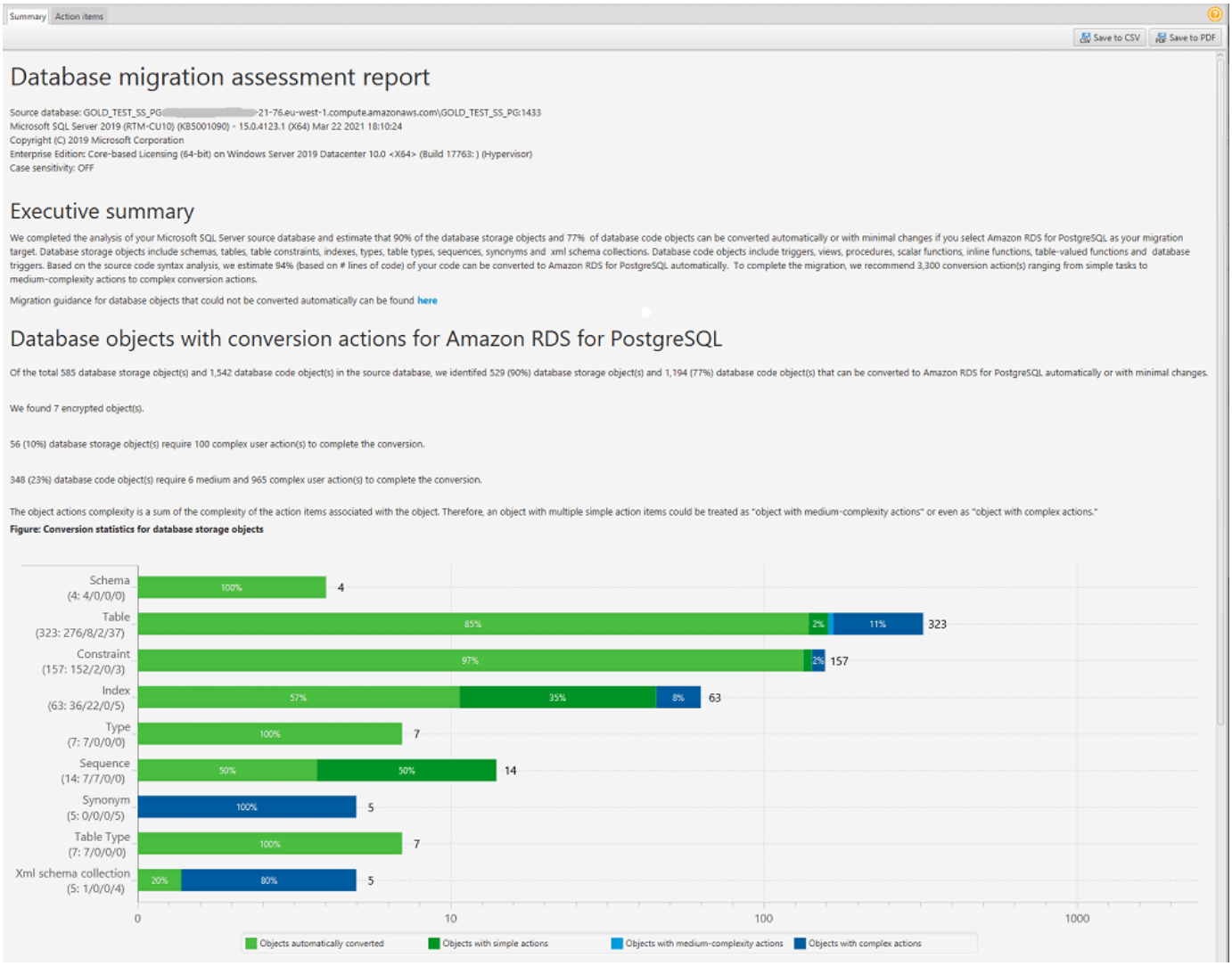


6. Wählen Sie die Registerkarte Summary aus.

   Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es wird sowohl die Anzahl der automatisch konvertierten Elemente angezeigt als auch die Anzahl der Elemente, die nicht automatisch konvertiert werden konnten. Die Übersicht enthält auch eine Schätzung der Zeit, die es dauert, Schemata in Ihrer Ziel-DB-Instance zu erstellen, die denen in Ihrer Quelldatenbank gleichen.

   Der Abschnitt Lizenzbewertung und Cloud-Support enthält Informationen zum Verschieben Ihres vorhandenen lokalen Datenbankschemas auf eine Amazon RDS-DB-Instance, auf der dieselbe Engine ausgeführt wird. Wenn Sie beispielsweise die Lizenztypen ändern möchten, erfahren Sie in diesem Abschnitt des Berichts, welche Funktionen aus Ihrer aktuellen Datenbank entfernt werden müssen.

   Ein Beispiel für eine Zusammenfassung eines Bewertungsberichts wird im Folgenden gezeigt.



7. Wählen Sie die Registerkarte Summary und anschließend Save to PDF aus. Der Bewertungsbericht zur Datenbankmigration wird als PDF-Datei gespeichert. Die PDF-Datei enthält sowohl die Zusammenfassung als auch Informationen zu den Aktionselementen.

   Sie können auch In CSV speichern wählen, um den Bericht als CSV-Datei zu speichern. Wenn Sie diese Option wählen, werden drei CSV-Dateien AWS SCT erstellt. Diese Dateien enthalten die folgenden Informationen:

   - Eine Liste von Konvertierungsaktionselementen mit empfohlenen Aktionen.
   - Eine Zusammenfassung der Konversionsmaßnahmen mit einer Schätzung des Aufwands, der erforderlich ist, um ein Ereignis der Aktion zu konvertieren.
   - Eine Zusammenfassung mit einer Reihe von Aktionspunkten, die nach der voraussichtlichen Konvertierungszeit kategorisiert sind.

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 identifed 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.

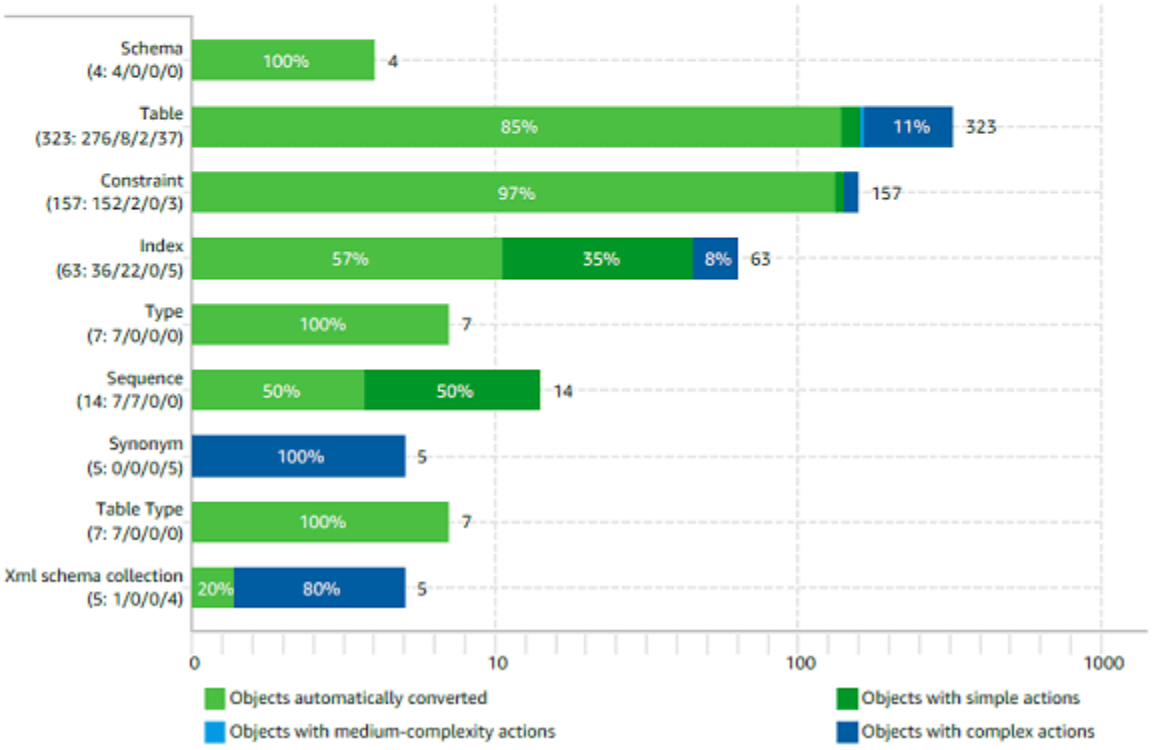
We found 7 encrypted object(s).

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

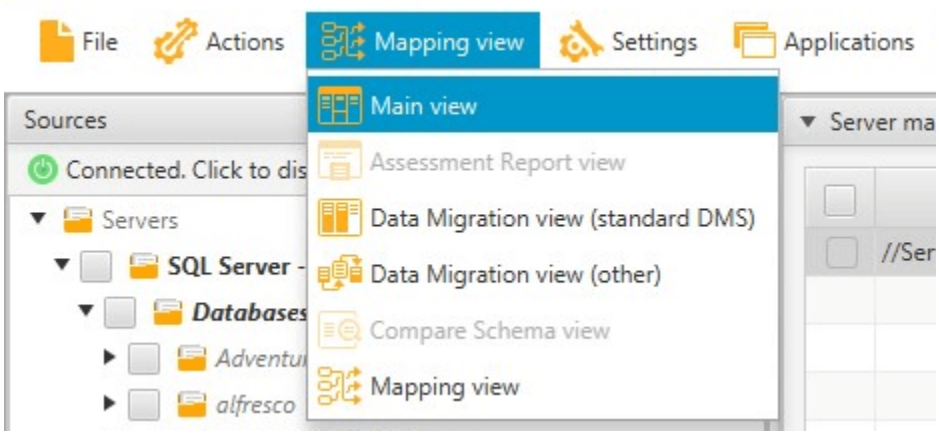


Schemas konvertieren in AWS Schema Conversion Tool

Nachdem Sie Ihrem Projekt Quell- und Zieldatenbanken hinzugefügt und Zuordnungsregeln erstellt haben, können Sie Ihre Quelldatenbankschemas konvertieren. Führen Sie die folgenden Schritte durch, um ein Schema zu konvertieren.

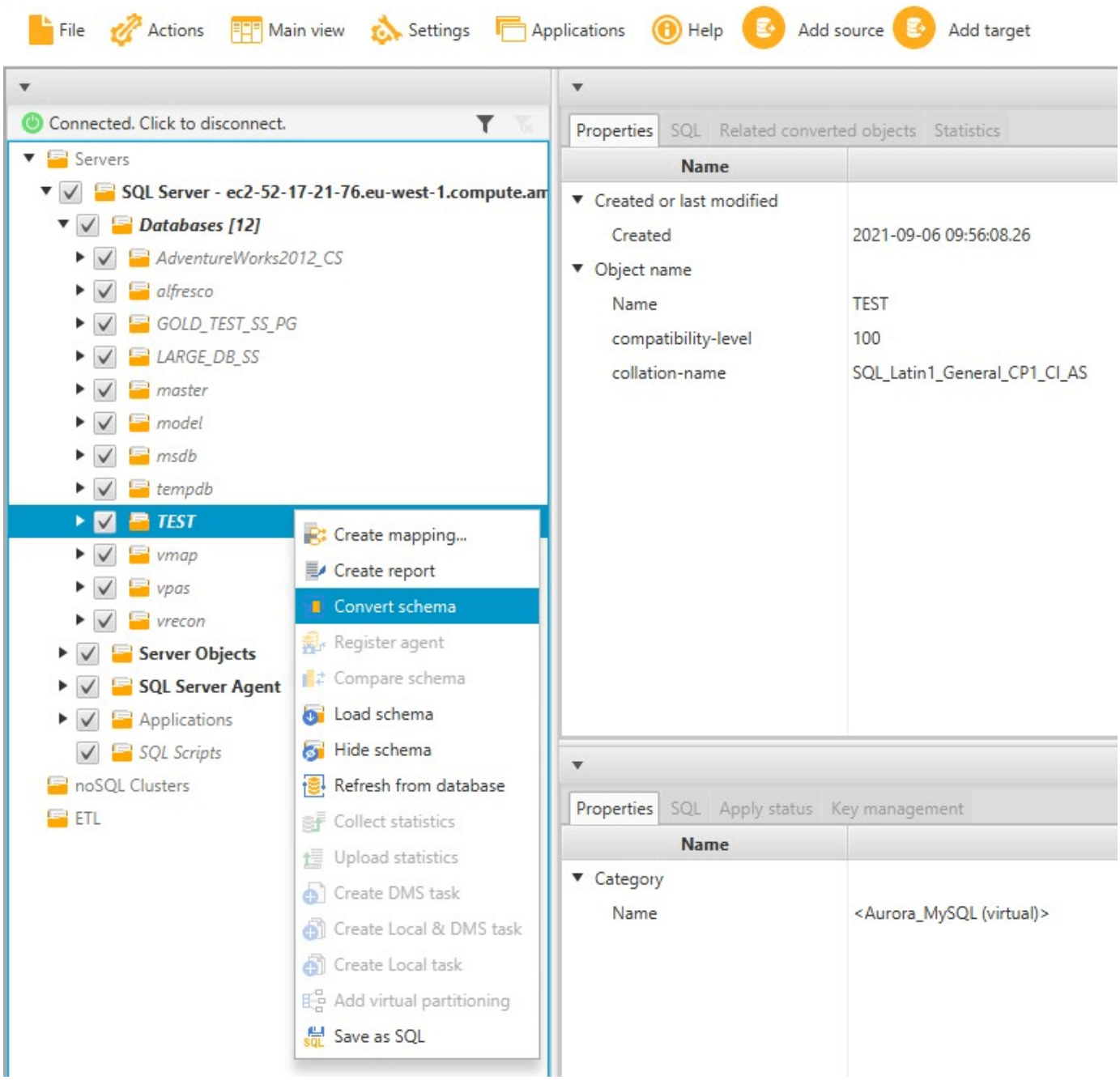
Um Ihr Schema zu konvertieren

1. Wählen Sie „Ansicht“ und anschließend „Hauptansicht“.



2. Aktivieren Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, das Kontrollkästchen für den Namen des zu konvertierenden Objekts. Wählen Sie als Nächstes dieses Objekt aus. AWS SCT hebt den Objektnamen blau hervor. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie Schema konvertieren.

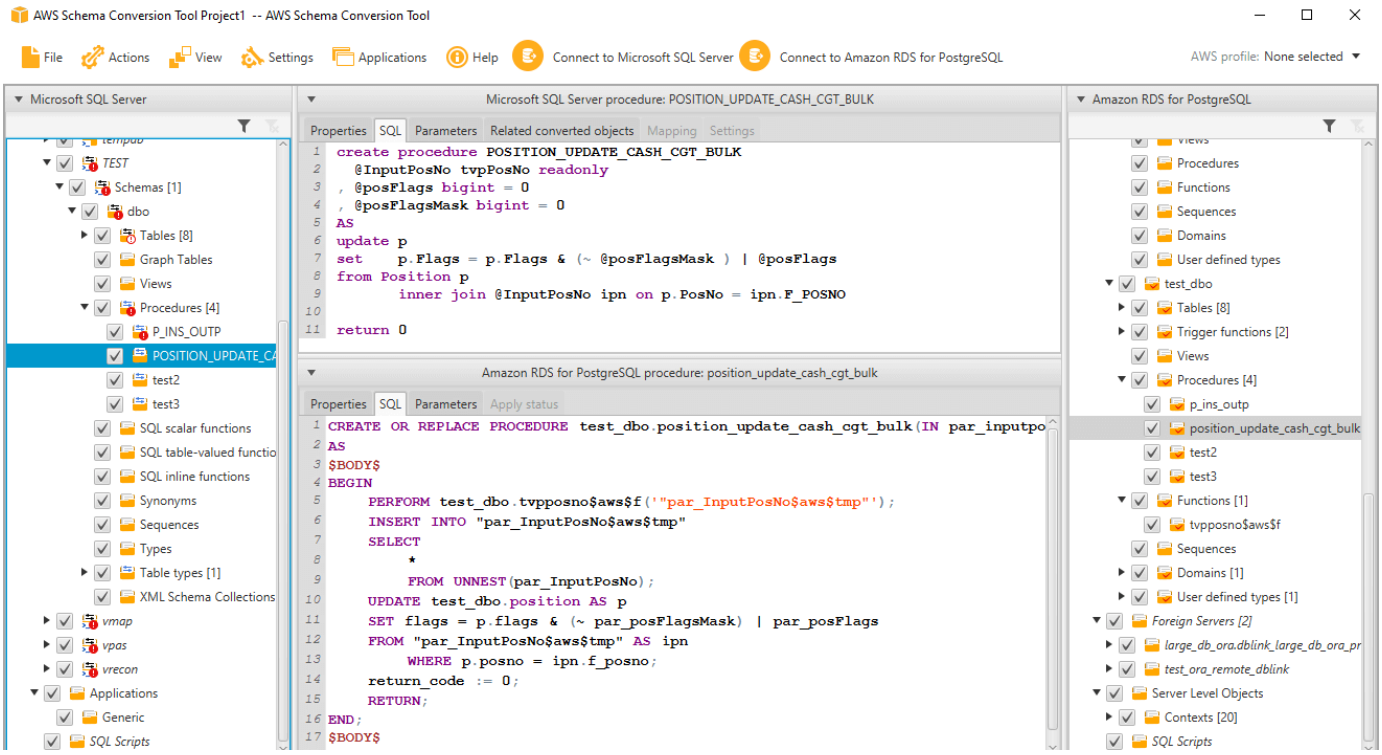
   Um mehrere Datenbankobjekte zu konvertieren, aktivieren Sie die Kontrollkästchen für alle Objekte. Wählen Sie als Nächstes den übergeordneten Knoten aus. Bei Tabellen ist der übergeordnete Knoten beispielsweise Tabellen. Stellen Sie sicher, dass dadurch der Name des übergeordneten Knotens blau AWS SCT hervorgehoben wird. Öffnen Sie das Kontextmenü (Rechtsklick) für den übergeordneten Knoten und wählen Sie Schema konvertieren.



3. Wenn Sie mit der Konvertierung des Schemas AWS SCT fertig sind, können Sie das vorgeschlagene Schema im Bereich auf der rechten Seite Ihres Projekts anzeigen.

   Zu diesem Zeitpunkt wird kein Schema auf Ihre Zieldatenbankinstanz angewendet. Das geplante Schema ist Teil Ihres Projekts. Wenn Sie ein konvertiertes Schemaelement auswählen, können Sie den geplanten Schemabefehl im Bereich unten in der Mitte für Ihre Zieldatenbank-Instance sehen.

Sie können das Schema in diesem Fenster bearbeiten. Das bearbeitete Schema wird als Teil Ihres Projekts gespeichert und in die Zieldatenbankinstanz geschrieben, wenn Sie Ihr konvertiertes Schema anwenden möchten.



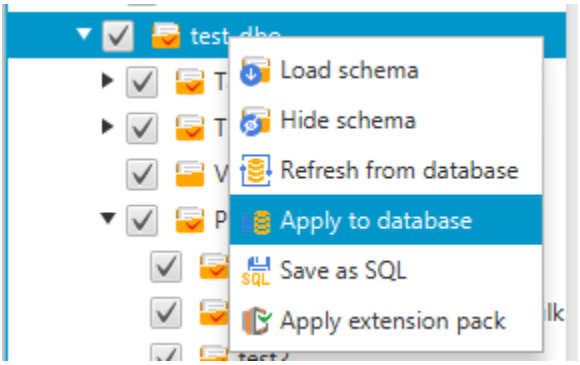
Anwenden der konvertierten Schemas in AWS Schema Conversion Tool

Sie können das konvertierte Datenbankschema auf Ihre DB-Ziel-Instance anwenden. Nachdem das Schema auf Ihre Ziel-DB-Instance angewendet wurde, können Sie das es basierend auf den Aktionselementen im Bewertungsbericht zur Datenbankmigration aktualisieren.

> **Warning**
>
> Das folgende Verfahren überschreibt das bestehende Zielschema. Achten Sie darauf, Schemas nicht ungewollt zu überschreiben. Achten Sie darauf, Schemas in Ihrer Ziel-DB-Instance, die Sie bereits geändert haben, nicht zu überschreiben, da Sie sonst diese Änderungen überschreiben.

Um das konvertierte Datenbankschema auf Ihre Zieldatenbank-Instance anzuwenden

1. Wählen Sie oben im rechten Bereich Ihres Projekts die Option Mit dem Server verbinden, um eine Verbindung zu Ihrer Zieldatenbank herzustellen. Wenn Sie mit Ihrer Zieldatenbank verbunden sind, überspringen Sie diesen Schritt.
2. Wählen Sie das Schema-Element im rechten Bereich Ihres Projekt aus, der das geplante Schema für Ihre Ziel-DB-Instance anzeigt.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schema-Element und wählen Sie Apply to database aus.



   Das konvertierte Schema wird auf Ihre Ziel-DB-Instance angewendet.

Profile verwalten in AWS Schema Conversion Tool

Sie können Ihre AWS Anmeldeinformationen in speichern AWS SCT. AWS SCT verwendet Ihre Anmeldeinformationen, wenn Sie Funktionen verwenden, die in AWS Dienste integriert werden. AWS SCT Integriert sich beispielsweise in Amazon S3 AWS Lambda, Amazon Relational Database Service (Amazon RDS) und AWS Database Migration Service (AWS DMS).

AWS SCT fragt Sie nach Ihren AWS Anmeldeinformationen, wenn Sie auf eine Funktion zugreifen, für die diese erforderlich sind. Sie können die Anmeldeinformationen in den globalen Anwendungseinstellungen speichern. Wenn Sie nach Ihren Anmeldeinformationen AWS SCT gefragt werden, können Sie die gespeicherten Anmeldeinformationen auswählen.

Sie können verschiedene Gruppen von AWS Anmeldeinformationen in den globalen Anwendungseinstellungen speichern. Sie können beispielsweise einen Anmeldeinformationssatz, den Sie in Testszenarien nutzen, und einen weiteren Anmeldeinformationssatz, den Sie in Produktionsszenarien verwenden, speichern. Sie können auch unterschiedliche Anmeldeinformationen für verschiedene AWS-Region s speichern.

AWS Anmeldeinformationen speichern

Gehen Sie wie folgt vor, um AWS Anmeldeinformationen global zu speichern.

Um AWS Anmeldeinformationen zu speichern

1. Starten Sie das AWS Schema Conversion Tool.
2. Öffnen Sie das Einstellungsmenü und wählen Sie dann Allgemeine Einstellungen. Das Dialogfeld Global settings wird angezeigt.
3. Wählen Sie AWS Dienstprofile und dann Neues AWS Dienstprofil hinzufügen aus.
4. Geben Sie Ihre AWS Informationen wie folgt ein.

AWS SCT Option	Aktion
Profilname	Geben Sie einen Namen für Ihr Profil ein.
AWS Zugriffsschlüssel	Geben Sie Ihren AWS Zugangsschlüssel ein.
AWS geheimer Schlüssel	Geben Sie Ihren AWS geheimen Zugangsschlüssel ein. Weitere Informationen zu AWS Zugriffsschlüsseln finden Sie unter Verwaltung von Zugriffsschlüsseln im IAM-Benutzerhandbuch.
Region	Wählen Sie das AWS-Region für Ihr Profil aus.
Amazon S3 S3-Bucket-Ordner	Wählen Sie den Amazon S3 S3-Bucket für Ihr Profil aus. Sie müssen nur dann einen Bucket angeben, wenn Sie eine Funktion verwenden , die eine Verbindung zu Amazon S3 herstellt. Weitere Informationen zu den erforderlichen Rechten finden Sie unterBerechtigungen für die Verwendung des AWS Dienstprofils.

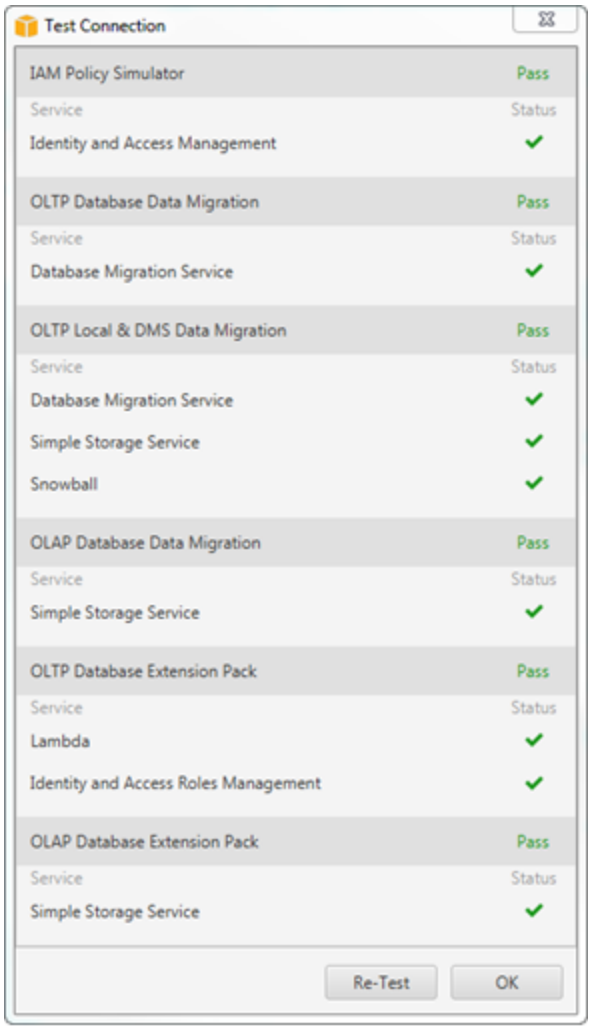
Wählen Sie Use FIPS endpoint for S3, wenn Sie die Sicherheitsanforderungen für die Federal Information Processing Standard (FIPS) erfüllen müssen. FIPS-Endpunkte sind in den folgenden AWS Regionen verfügbar:

- Region USA Ost (Nord-Virginia)
- Region USA Ost (Ohio)

- Region USA West (Nordkalifornien)
- Region USA West (Oregon)

5. Wählen Sie Verbindung testen, um zu überprüfen, ob Ihre Anmeldeinformationen korrekt und aktiv sind.

   Das Dialogfeld Verbindung testen wird angezeigt. Für jeden Service, der mit Ihrem Profil verknüpft ist, wird der Status angegeben. Die Option Pass gibt an, dass das Profil erfolgreich auf den Service zugreifen kann.



6. Nachdem Sie Ihr Profil konfiguriert haben, wählen Sie Save, um Ihr Profil zu speichern, oder Cancel, um die Änderungen zu verwerfen.
7. Wählen Sie OK, um das Dialogfeld Allgemeine Einstellungen zu schließen.

Das Standardprofil für ein Projekt festlegen

Sie können das Standardprofil für ein AWS SCT Projekt festlegen. Dadurch werden die im Profil gespeicherten AWS Anmeldeinformationen dem Projekt zugeordnet. Öffnen Sie das Projekt und führen Sie die folgenden Schritte aus, um das Standardprofil festzulegen.

So legen Sie das Standardprofil für ein Projekt fest

1. Starten Sie das AWS Schema Conversion Tool und erstellen Sie ein neues Projekt.
2. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen aus. Das Dialogfeld mit den Projekteinstellungen wird angezeigt.
3. Wählen Sie die Registerkarte Projektumgebung.
4. Wählen Sie Neues AWS Dienstprofil hinzufügen, um ein neues Profil hinzuzufügen. Wählen Sie dann für das AWS Dienstprofil das Profil aus, das Sie dem Projekt zuordnen möchten.
5. Wählen Sie OK, um das Dialogfeld mit den Projekteinstellungen zu schließen. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.

Berechtigungen für die Verwendung des AWS Dienstprofils

Für den Zugriff auf Ihren Amazon S3 S3-Bucket von Ihrem AWS Serviceprofil aus sind die folgenden Berechtigungen erforderlich:

- `s3:PutObject`— um Objekte zu Ihrem Amazon S3 S3-Bucket hinzuzufügen.
- `s3:DeleteObject`— um die Null-Version eines Objekts zu entfernen und eine Löschmarkierung einzufügen, die zur aktuellen Version des Objekts wird.
- `s3:ListBucket`— um bis zu 1.000 Objekte aus Ihrem Amazon S3 S3-Bucket zurückzugeben.
- `s3:GetObject`— um Objekte aus Ihrem Amazon S3 S3-Bucket abzurufen.

Das folgende Codebeispiel zeigt Ihnen, wie Sie Ihrem Benutzer diese Berechtigungen gewähren können.

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

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

Konfiguration AWS Secrets Manager in der AWS Schema Conversion Tool

AWS SCT kann Datenbankanmeldedaten verwenden, die Sie in speichern AWS Secrets Manager. Sie können alle Werte in das Datenbankverbindungsdialogfeld von Secrets Manager eingeben. Um Secrets Manager zu verwenden, stellen Sie sicher, dass Sie AWS Profile in der speichern AWS Schema Conversion Tool.

Weitere Informationen zur Verwendung AWS Secrets Manager finden Sie unter Was ist AWS Secrets Manager? im AWS Secrets Manager Benutzerhandbuch. Weitere Informationen zum Speichern von AWS Profilen finden Sie unterProfile verwalten in AWS Schema Conversion Tool.

Um Datenbankanmeldedaten von Secrets Manager abzurufen

1. Starten Sie das AWS Schema Conversion Tool und erstellen Sie ein neues Projekt.
2. Wählen Sie Quelle hinzufügen oder Ziel hinzufügen, um Ihrem Projekt eine neue Datenbank hinzuzufügen.
3. Wählen Sie eine Datenbankplattform und dann Weiter.
4. Wählen Sie für AWS Secret das Secret aus, das Sie verwenden möchten.
5. Wählen Sie Populate aus. AWS SCT Füllt dann alle Werte in das Datenbankverbindungsdialogfeld ein.
6. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Datenbank hergestellt werden AWS SCT kann.
7. Wählen Sie Connect, um eine Verbindung zu Ihrer Datenbank herzustellen.

AWS SCT unterstützt Secrets mit der folgenden Struktur.

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

In dieser Struktur sind die `password` Werte `username` und erforderlich, und alle anderen Werte sind optional. Stellen Sie sicher, dass die Werte, die Sie in Secrets Manager speichern, alle Datenbankanmeldedaten enthalten.

Speichern von Passwörtern im AWS Schema Conversion Tool

Sie können ein Datenbankkennwort oder ein SSL-Zertifikat im AWS SCT Cache speichern. Um ein Passwort zu speichern, wählen Sie Store Password, (Passwort speichern) aus, wenn Sie eine Verbindung erstellen.

Das Passwort wird unter Verwendung des zufällig generierten Token in der Datei `seed.dat` verschlüsselt. Das Passwort wird dann mit dem Benutzernamen in der Cache-Datei gespeichert. Ist die Datei `seed.dat` nicht mehr vorhanden oder beschädigt, ist das Datenbankpasswort möglicherweise nicht richtig verschlüsselt. In diesem Fall kann keine Verbindung hergestellt werden.

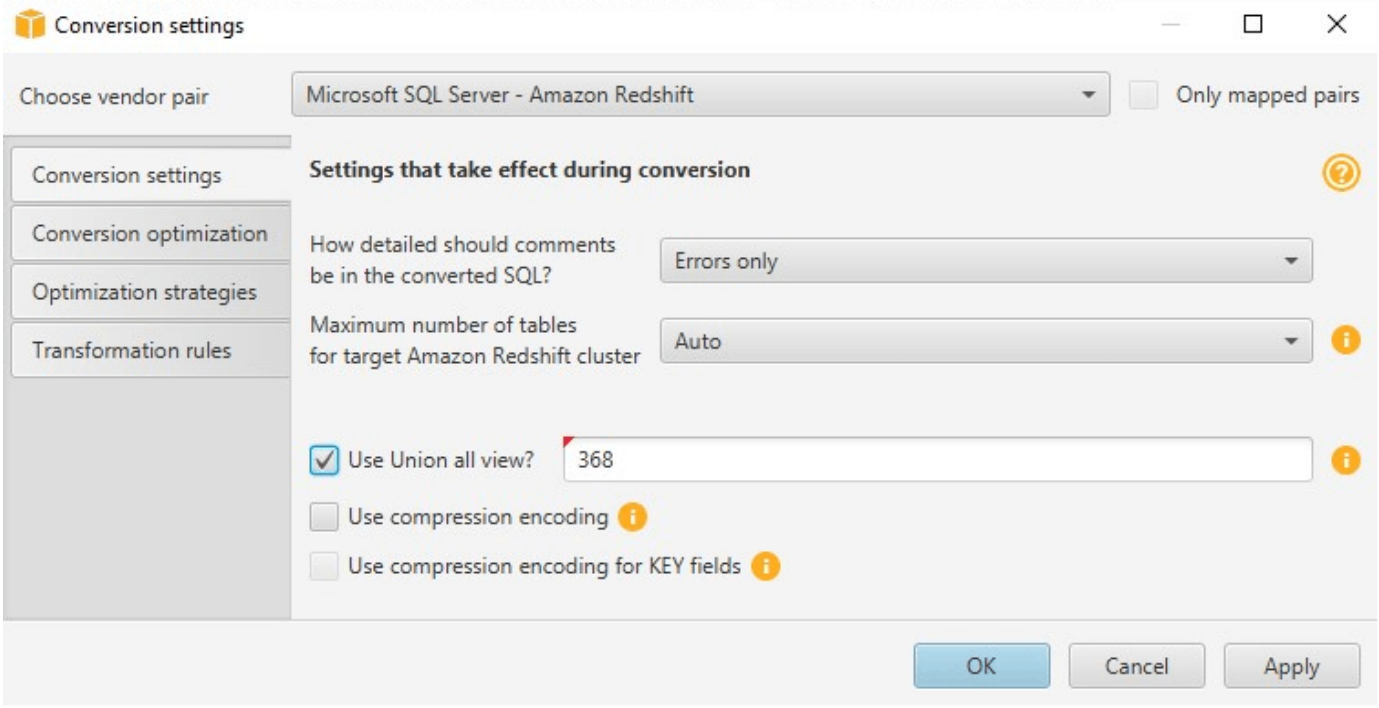
Die UNION ALL-Ansicht wird erstellt in AWS Schema Conversion Tool

Wenn eine Quelltabelle partitioniert ist, werden n Zieltabellen AWS SCT erstellt, wobei n die Anzahl der Partitionen in der Quelltabelle ist. AWS SCT erstellt eine UNION ALL-Ansicht über den Zieltabellen, um die Quelltabelle darzustellen. Wenn Sie einen AWS SCT Datenextraktor verwenden, um Ihre Daten zu migrieren, werden die Quelltabellenpartitionen parallel durch separate Unteraufgaben extrahiert und geladen.

So verwenden Sie Union Alle View für ein Projekt

1. Fangen Sie an. AWS SCT Erstellen Sie ein neues Projekt oder öffnen Sie ein vorhandenes AWS SCT Projekt.

2. Wählen Sie im Menü Einstellungen die Option Konvertierungseinstellungen aus.
3. Wählen Sie ein Paar OLAP-Datenbanken aus der Liste oben aus.
4. Union-Ansicht für alle verwenden aktivieren?



5. Wählen Sie OK, um die Einstellungen zu speichern und das Dialogfeld mit den Konvertierungseinstellungen zu schließen.

Verwenden von Tastenkombinationen im AWS Schema Conversion Tool

Im Folgenden finden Sie die Tastenkombinationen, die Sie mit AWS SCT verwenden können.

Tastaturkürzel	Beschreibung
Strg+N	Erstellt ein neues Projekt.
Strg+O	Öffnet ein vorhandenes Projekt.
Strg+S	Speichert ein geöffnetes Projekt.
Strg+W	Erstellt ein neues Projekt mithilfe des Assistenten.

Tastaturkürzel	Beschreibung
Strg+M	Erstellen Sie ein neues Multiserver-Assessment.
Strg+L	Fügen Sie eine neue Quelldatenbank hinzu.
Strg+R	Fügen Sie eine neue Zieldatenbank hinzu.
Strg+F4	Schließt ein geöffnetes Projekt.
F1	Öffnen Sie das AWS SCT Benutzerhandbuch.

Erste Schritte mit AWS Schema Conversion Tool

Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um das Schema für eine Quelldatenbank zu konvertieren. Die Quelldatenbank kann eine selbstverwaltete Engine sein, die lokal oder auf einer EC2 Amazon-Instance ausgeführt wird. Sie können Ihr Quellschema in ein Schema für jede unterstützte Datenbank konvertieren, die von gehostet wird. AWS Die AWS SCT Anwendung bietet eine projektbasierte Benutzeroberfläche.

Fast alle Arbeiten, mit denen Sie arbeiten, AWS SCT beginnen mit den folgenden Schritten:

1. Installiere AWS SCT. Weitere Informationen finden Sie unter Installation und Konfiguration AWS Schema Conversion Tool.
2. Installieren Sie bei Bedarf einen AWS SCT Agenten. AWS SCT Agenten sind nur für bestimmte Migrationsszenarien erforderlich, z. B. zwischen heterogenen Quellen und Zielen. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.
3. Machen Sie sich mit der Benutzeroberfläche von AWS SCT vertraut. Weitere Informationen finden Sie unter Navigieren in der Benutzeroberfläche des AWS SCT.
4. Erstellen Sie ein AWS SCT Projekt. Stellen Sie eine Verbindung mit Ihren Quell- und Zieldatenbanken her. Weitere Information zur Herstellung einer Verbindung zu Ihrer Quelldatenbank finden Sie unter Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool.
5. Erstellen Sie Zuordnungsregeln. Weitere Informationen zu Zuordnungsregeln finden Sie unterZuordnung von Datentypen in der AWS Schema Conversion Tool.
6. Führen Sie den Database Migration Assessment Report (Bewertungsbericht zur Datenbankmigration) aus und zeigen Sie ihn an. Weitere Informationen zum Bewertungsbericht finden Sie unter Den Bewertungsbericht anzeigen in AWS Schema Conversion Tool.
7. Wandeln Sie die Quelldatenbank Schemas um. Es gibt mehrere Aspekte bei der Umwandlung, die Sie berücksichtigen müssen, z. B. was mit Elementen geschehen soll, die nicht umgewandelt werden, und wie Elemente abgebildet werden, die auf eine bestimmte Weise umgewandelt werden sollen. Weitere Informationen zur Umwandlung eines Quellschemas finden Sie unter Datenbankschemas konvertieren in AWS Schema Conversion Tool.

   Wenn Sie ein Data Warehouse-Schema umwandeln, müssen vor der Umwandlung ebenfalls einige Dinge berücksichtigt werden. Weitere Informationen finden Sie unter Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT.

8. Anwenden der Schemaumwandlung auf Ihr Ziel. Weitere Informationen zur Anwendung einer Quellschemaumwandlung finden Sie unter Anwenden der konvertierten Schemas.
9. Sie können es auch verwenden AWS SCT , um gespeicherte SQL-Prozeduren und anderen Anwendungscode zu konvertieren. Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren mit AWS SCT

Sie können es auch verwenden AWS SCT , um Ihre Daten von einer Quelldatenbank in eine von Amazon verwaltete Datenbank zu migrieren. Beispiele finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.

Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool

AWS Schema Conversion Tool (AWS SCT) kann Schemas aus den folgenden Quelldatenbanken und Data Warehouses in eine Zieldatenbank oder ein Data Warehouse konvertieren. Informationen zu Berechtigungen, Verbindungen und Informationen darüber, was für die Verwendung mit der Zieldatenbank oder dem Data Warehouse konvertiert werden AWS SCT kann, finden Sie in den Details in den folgenden Themen.

Informationen zur Verschlüsselung

Verbindung zu verschlüsseltem Amazon RDS und Aurora herstellen

Datenbankquellen

- Verbindung zu Apache Cassandra herstellen
- Verbindung zu Azure SQL herstellen
- Verbindung zu IBM DB2 für z/OS herstellen
- IBM Db2 LUW-Datenbanken
- Verwenden von MySQL als Quelle
- Oracle-Datenbanken
- PostgreSQL-Datenbanken
- SAP-Datenbanken
- SQL Server-Datenbanken

Data Warehouse-Quellen

- Amazon Redshift
- Azure Synapse Analytics als Quelle
- BigQuery als Quelle
- Greenplum-Datenbanken
- Netezza-Datenbanken
- Oracle Data Warehouse
- Snowflake

- SQL Server-Data Warehouses
- Teradata-Datenbanken
- Vertica-Datenbanken

Große Datenquellen

- Verbindung zu Apache Hadoop herstellen
- Verbindung zu Apache Oozie herstellen

Herstellen einer Verbindung zu verschlüsselten Amazon Relational Database Service- und Amazon Aurora Aurora-Datenbanken mit dem AWS Schema Conversion Tool

Um verschlüsselte Verbindungen zu Amazon RDS- oder Amazon Aurora Aurora-Datenbanken von einer Anwendung aus zu öffnen, müssen Sie AWS Stammzertifikate in eine Form von Schlüsselspeicher importieren. Sie können die Stammzertifikate AWS unter Verwenden von SSL/TLS zum Verschlüsseln einer Verbindung zu einer DB-Instance im Amazon RDS-Benutzerhandbuch herunterladen.

Zwei Optionen sind verfügbar: ein Stammzertifikat, das für alle AWS Regionen funktioniert, und ein Zertifikatspaket, das sowohl die alten als auch die neuen Stammzertifikate enthält.

Je nachdem, welches Sie verwenden möchten, folgen Sie den Schritten in einem der beiden folgenden Verfahren.

Um das Zertifikat oder die Zertifikate in den Windows-Systemspeicher zu importieren

1. Laden Sie ein oder mehrere Zertifikate aus einer der folgenden Quellen herunter:

   Informationen zum Herunterladen von Zertifikaten finden Sie unter Verwenden von SSL/TLS zur Verschlüsselung einer Verbindung zu einer DB-Instance im Amazon RDS-Benutzerhandbuch.
2. Geben Sie in Ihrem Windows-Suchfenster ein. **`Manage computer certificates`** Wenn Sie gefragt werden, ob die Anwendung Änderungen an Ihrem Computer vornehmen soll, wählen Sie Ja.
3. Wenn das Fenster mit den Zertifikaten geöffnet wird, erweitern Sie bei Bedarf die Option Zertifikate — Lokaler Computer, sodass Sie die Liste der Zertifikate sehen können. Öffnen Sie

das Kontextmenü (Rechtsklick) für Vertrauenswürdige Stammzertifizierungsstellen und wählen Sie dann Alle Aufgaben, Import aus.

4. Wählen Sie Weiter und dann Durchsuchen und suchen Sie die `*.pem` Datei, die Sie in Schritt 1 heruntergeladen haben. Wählen Sie Öffnen, um die Zertifikatsdatei auszuwählen, klicken Sie auf Weiter und dann auf Fertig stellen.

> **Note**
>
> Um die Datei zu finden, ändern Sie den Dateityp im Suchfenster in Alle Dateien (*.*), da es sich nicht um eine standardmäßige Zertifikatserweiterung `.pem` handelt.

5. Erweitern Sie in der Microsoft Management Console den Bereich Zertifikate. Erweitern Sie dann Vertrauenswürdige Stammzertifizierungsstellen, wählen Sie Zertifikate aus und suchen Sie nach dem Zertifikat, um zu bestätigen, dass es existiert. Der Name des Zertifikats beginnt mit`Amazon RDS`.
6. Starten Sie Ihren Computer neu.

Um das Zertifikat oder die Zertifikate in Java zu importieren KeyStore

1. Laden Sie das Zertifikat oder die Zertifikate aus einer der folgenden Quellen herunter:

   Informationen zum Herunterladen von Zertifikaten finden Sie unter Verwenden von SSL/TLS zur Verschlüsselung einer Verbindung zu einer DB-Instance im Amazon RDS-Benutzerhandbuch.
2. Wenn Sie das Zertifikatspaket heruntergeladen haben, teilen Sie es in einzelne Zertifikatsdateien auf. Platzieren Sie dazu jeden Zertifikatsblock, beginnend mit `-----BEGIN CERTIFICATE-----` und endend mit, `-----END CERTIFICATE-----` in separate `*.pem` Dateien. Nachdem Sie für jedes Zertifikat eine separate `*.pem` Datei erstellt haben, können Sie die Zertifikatspaketdatei problemlos entfernen.
3. Öffnen Sie ein Befehlsfenster oder eine Terminalsitzung in dem Verzeichnis, in das Sie das Zertifikat heruntergeladen haben, und führen Sie den folgenden Befehl für jede `*.pem` Datei aus, die Sie im vorherigen Schritt erstellt haben.

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

Example

Im folgenden Beispiel wird davon ausgegangen, dass Sie die `eu-west-1-bundle.pem` Datei heruntergeladen haben.

```
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. Fügen Sie den Keystore als Trust Store in hinzu. AWS SCTWählen Sie dazu im Hauptmenü Einstellungen, Allgemeine Einstellungen, Sicherheit, Vertrauensspeicher und dann Bestehenden Vertrauensspeicher auswählen aus.

   Nachdem Sie den Trust Store hinzugefügt haben, können Sie ihn verwenden, um eine SSL-fähige Verbindung zu konfigurieren, wenn Sie eine AWS SCT Verbindung zur Datenbank herstellen. Wählen AWS SCT Sie im Dialogfeld Mit Datenbank verbinden die Option SSL verwenden und wählen Sie den zuvor eingegebenen Vertrauensspeicher aus.

Herstellen einer Verbindung zu Apache Cassandra-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schlüsselräume von Apache Cassandra nach Amazon DynamoDB zu konvertieren.

Verbindung zu Apache Cassandra als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Apache Cassandra-Quelldatenbank herzustellen. AWS Schema Conversion Tool

Um eine Verbindung zu einer Apache Cassandra-Quelldatenbank herzustellen

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen.
2. Wählen Sie Cassandra und dann Weiter.

Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unter Konfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Apache Cassandra-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie nach dem Passwort gefragt, um bei

Parameter	Aktion
	Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustel len.
Use SSL	Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informati onen ein: • Trust Store: Der zu verwendende Trust Store. • Schlüsselspeicher: Der zu verwendende Schlüssel speicher.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Herstellen einer Verbindung zu Apache Hadoop-Datenbanken mit dem AWS Schema Conversion Tool

Sie können die AWS SCT Befehlszeilenschnittstelle (CLI) verwenden, um von Apache Hadoop zu Amazon EMR zu migrieren. AWS SCT verwendet Ihren Amazon S3 S3-Bucket als temporären Speicher für Ihre Daten während der Migration.

AWS SCT unterstützt als Quelle Apache Hadoop Version 2.2.0 und höher. AWS SCT Unterstützt auch Apache Hive Version 0.13.0 und höher.

AWS SCT unterstützt als Ziel Amazon EMR Version 6.3.0 und höher. AWS SCT Unterstützt als Zielversion auch Apache Hadoop Version 2.6.0 und höher und Apache Hive Version 0.13.0 und höher.

Themen

- Voraussetzungen für die Verwendung von Apache Hadoop als Quelle
- Berechtigungen für die Verwendung von Hive als Quelle
- Berechtigungen für die Verwendung von HDFS als Quelle
- Berechtigungen für die Verwendung von HDFS als Ziel
- Verbindung zu Apache Hadoop als Quelle herstellen
- Stellen Sie eine Verbindung zu Ihren Hive- und HDFS-Quelldiensten her
- Verbindung zu Amazon EMR als Ziel herstellen

Voraussetzungen für die Verwendung von Apache Hadoop als Quelle

Die folgenden Voraussetzungen sind erforderlich, um über die AWS SCT CLI eine Verbindung zu Apache Hadoop herzustellen.

- Erstellen Sie einen Amazon S3 S3-Bucket, um Daten während der Migration zu speichern. Anschließend können Sie Daten auf Amazon EMR HDFS kopieren oder Amazon S3 als Datenspeicher für Ihre Hadoop-Workloads verwenden. Weitere Informationen finden Sie unter Erstellen eines Buckets im Amazon S3 S3-Benutzerhandbuch.
- Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle mit der `AmazonS3FullAccess` Richtlinie. AWS SCT verwendet diese IAM-Rolle, um auf Ihren Amazon S3 S3-Bucket zuzugreifen.
- Notieren Sie sich Ihren AWS geheimen Schlüssel und Ihren AWS geheimen Zugriffsschlüssel. Weitere Informationen zu AWS Zugriffsschlüsseln finden Sie unter Verwaltung von Zugriffsschlüsseln im IAM-Benutzerhandbuch.
- Erstellen und konfigurieren Sie einen Amazon EMR-Zielcluster. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EMR im Amazon EMR Management Guide.
- Installieren Sie das `distcp` Hilfsprogramm auf Ihrem Apache Hadoop-Quellcluster. Installieren Sie das `s3-dist-cp` Hilfsprogramm außerdem auf Ihrem Amazon EMR-Zielcluster. Stellen Sie sicher, dass Ihre Datenbankbenutzer berechtigt sind, diese Dienstprogramme auszuführen.

- Konfigurieren Sie die `core-site.xml` Datei in Ihrem Quell-Hadoop-Cluster so, dass sie das S3A-Protokoll verwendet. Setzen Sie dazu den `fs.s3a.aws.credentials.provider` Parameter auf einen der folgenden Werte.
  - `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`

  Sie können der `core-site.xml` Datei das folgende Codebeispiel hinzufügen.

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

  Das vorherige Beispiel zeigt eine der vier Optionen aus der vorherigen Optionsliste. Wenn Sie den `fs.s3a.aws.credentials.provider` Parameter nicht in der `core-site.xml` Datei festlegen, wird AWS SCT der Anbieter automatisch ausgewählt.

Berechtigungen für die Verwendung von Hive als Quelle

Die für einen Hive-Quellbenutzer erforderlichen Berechtigungen lauten wie folgt:

- `READ`Zugriff auf die Quelldatenordner und auf den Amazon S3-Quell-Bucket
- `READ+WRITE`Zugriff auf die Zwischen- und Ziel-Amazon S3-Buckets

Um die Migrationsgeschwindigkeit zu erhöhen, empfehlen wir, die Komprimierung für ACID-Transaktions-Quelltabellen durchzuführen.

Die für einen Amazon EMR Hive-Zielbenutzer erforderlichen Berechtigungen lauten wie folgt:

- `READ`Zugriff auf den Amazon S3-Ziel-Bucket
- `READ+WRITE`Zugriff auf den Amazon S3 S3-Zwischenbucket
- `READ+WRITE`Zugriff auf die HDFS-Zielordner

Berechtigungen für die Verwendung von HDFS als Quelle

Die für HDFS als Quelle erforderlichen Berechtigungen lauten wie folgt:

- `EXECUTE`für NameNode
- `EXECUTE+READ`für alle Quellordner und Dateien, die Sie in das Migrationsprojekt aufnehmen
- `READ+WRITE`für das `tmp` Verzeichnis im NameNode , um Spark-Jobs auszuführen und Dateien vor der Migration zu Amazon S3 zu speichern

In HDFS benötigen alle Operationen Traversal-Zugriff. Für den Traversalzugriff ist die `EXECUTE` Genehmigung für alle vorhandenen Komponenten des Pfads erforderlich, mit Ausnahme der letzten Pfadkomponente. Zum Beispiel muss Ihr Benutzer für jede Operation`/foo/bar/baz`, auf die zugegriffen wird, über `EXECUTE` Berechtigungen für `//foo`, und verfügen. `/foo/bar`

Das folgende Codebeispiel zeigt, wie Sie `EXECUTE+READ` Berechtigungen für Ihre Quellordner und -dateien sowie `READ+WRITE` Berechtigungen für das `tmp` Verzeichnis gewähren.

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

Berechtigungen für die Verwendung von HDFS als Ziel

Die für Amazon EMR HDFS als Ziel erforderlichen Berechtigungen lauten wie folgt:

- `EXECUTE`für den NameNode des Amazon EMR-Ziel-Clusters
- `READ+WRITE`für die HDFS-Zielordner, in denen Sie nach der Migration Daten speichern werden

Verbindung zu Apache Hadoop als Quelle herstellen

Sie können Apache Hadoop in AWS SCT Version 1.0.670 oder höher als Quelle verwenden. Sie können Hadoop-Cluster nur über die AWS SCT Befehlszeilenschnittstelle (CLI) zu Amazon EMR migrieren. Bevor Sie beginnen, machen Sie sich mit der Befehlszeilenschnittstelle von vertraut. AWS SCT Weitere Informationen finden Sie unter CLI-Referenz für AWS Schema Conversion Tool.

So stellen Sie in der CLI eine Verbindung zu Apache Hadoop her AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können die `HadoopMigrationTemplate.scts` Vorlage beispielsweise herunterladen und bearbeiten. Weitere Informationen finden Sie unter CLI-Szenarien abrufen.
2. Konfigurieren Sie die AWS SCT Anwendungseinstellungen wie den Speicherort des Treibers und den Protokollordner.

   Laden Sie den erforderlichen JDBC-Treiber herunter und geben Sie den Speicherort für die Datei an. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

   Das folgende Codebeispiel zeigt Ihnen, wie Sie den Pfad zum Apache Hive-Treiber hinzufügen. AWS SCT Speichert Protokolldateien im `c:\sct` Ordner, nachdem Sie dieses Codebeispiel ausgeführt haben.

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

   Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden.
3. Erstellen Sie ein neues AWS SCT Projekt.

   Im folgenden Codebeispiel wird das `hadoop_emr` Projekt im `c:\sct` Ordner erstellt.

   ```
   CreateProject
       -name: 'hadoop_emr'
       -directory: 'c:\sct'
   /
   ```
4. Fügen Sie dem Projekt Ihren Quell-Hadoop-Cluster hinzu.

   Verwenden Sie den `AddSourceCluster` Befehl, um eine Verbindung zum Quell-Hadoop-Cluster herzustellen. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben:`name`, `hostport`, und`user`. Andere Parameter sind optional.

Das folgende Codebeispiel fügt den Quell-Hadoop-Cluster hinzu. In diesem HADOOP_SOURCE Beispiel wird der Name des Quellclusters festgelegt. Verwenden Sie diesen Objektnamen, um Hive- und HDFS-Dienste zum Projekt hinzuzufügen und Zuordnungsregeln zu erstellen.

```
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'
/
```

Ersetzen Sie es im vorherigen Beispiel *hadoop_address* durch die IP-Adresse Ihres Hadoop-Clusters. Konfigurieren Sie bei Bedarf den Wert der Port-Option. Ersetzen Sie als Nächstes *hadoop_user* und *hadoop_password* durch den Namen Ihres Hadoop-Benutzers und das Passwort für diesen Benutzer. Geben Sie für *path\name* den Namen und den Pfad zur PEM-Datei für Ihren Hadoop-Quellcluster ein.

5. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihre Hive- und HDFS-Dienste hinzu.

Stellen Sie eine Verbindung zu Ihren Hive- und HDFS-Quelldiensten her

Sie können mit der CLI eine Verbindung zu Ihren Hive- und HDFS-Quelldiensten herstellen. AWS SCT Um eine Verbindung zu Apache Hive herzustellen, verwenden Sie den Hive-JDBC-Treiber Version 2.3.4 oder höher. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

AWS SCT stellt mit dem Cluster-Benutzer eine Verbindung zu Apache Hive her. hadoop Verwenden Sie dazu die AddSourceClusterHDFS Befehle AddSourceClusterHive und. Sie können einen der folgenden Ansätze verwenden.

- Erstellen Sie einen neuen SSH-Tunnel.

Geben Sie unter `createTunnel` den Wert **`true`** ein. Geben Sie für `host` die interne IP-Adresse Ihres Quell-Hive- oder HDFS-Dienstes ein. Geben Sie für `port` den Service-Port Ihres Hive- oder HDFS-Dienstes ein.

Geben Sie als Nächstes Ihre Hive- oder HDFS-Anmeldeinformationen für und ein. `user` `password` Weitere Informationen zu SSH-Tunneln finden Sie unter Einrichten eines SSH-Tunnels zum primären Knoten mithilfe der lokalen Portweiterleitung im Amazon EMR Management Guide.

- Verwenden Sie einen vorhandenen SSH-Tunnel.

  Geben Sie unter `host` den Wert **`localhost`** ein. Geben Sie für `port` den lokalen Port aus den SSH-Tunnelparametern ein.

- Stellen Sie direkt Connect zu Ihren Hive- und HDFS-Diensten her.

  Geben Sie für `host` die IP-Adresse oder den Hostnamen Ihres Quell-Hive- oder HDFS-Dienstes ein. Geben Sie für `port` den Service-Port Ihres Hive- oder HDFS-Dienstes ein. Geben Sie als Nächstes Ihre Hive- oder HDFS-Anmeldeinformationen für und ein. `user` `password`

So stellen Sie in der CLI eine Verbindung zu Hive und HDFS her AWS SCT

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihren Hadoop-Quellcluster enthält. Stellen Sie sicher, dass Sie den Namen des Hadoop-Clusters verwenden, den Sie im vorherigen Schritt definiert haben.
2. Fügen Sie dem Projekt Ihren Quell-Hive-Dienst hinzu.

   Verwenden Sie den `AddSourceClusterHive` Befehl, um eine Verbindung zum Quell-Hive-Dienst herzustellen. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben: `userpassword`,, `clustername`, und`port`. Andere Parameter sind optional.

   Das folgende Codebeispiel erstellt einen Tunnel für AWS SCT die Arbeit mit Ihrem Hive-Dienst. Dieser Quell-Hive-Dienst läuft auf demselben PC wie. AWS SCT In diesem Beispiel wird der `HADOOP_SOURCE` Quellcluster aus dem vorherigen Beispiel verwendet.

```
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'
/
```

Das folgende Codebeispiel stellt ohne Tunnel eine Verbindung zu Ihrem Hive-Dienst her.

```
AddSourceClusterHive
    -cluster: 'HADOOP_SOURCE'
    -name: 'HIVE_SOURCE'
    -host: 'hive_address'
    -port: 'hive_port'
    -user: 'hive_user'
    -password: 'hive_password'
/
```

Ersetzen Sie in den vorherigen Beispielen *hive_user* und *hive_password* durch den Namen Ihres Hive-Benutzers und das Passwort für diesen Benutzer.

Ersetzen Sie als Nächstes *hive_address* und *hive_port* durch die NameNode IP-Adresse und den Port Ihres Quell-Hadoop-Clusters.

Für *hive_remote_address* können Sie den Standardwert `127.0.0.1` oder die NameNode IP-Adresse Ihres Quell-Hive-Dienstes verwenden.

3. Fügen Sie dem Projekt Ihren Quell-HDFS-Dienst hinzu.

   Verwenden Sie den `AddSourceClusterHDFS` Befehl, um eine Verbindung zum Quell-HDFS-Dienst herzustellen. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben: `userpassword`,, `clustername`, und`port`. Andere Parameter sind optional.

   Stellen Sie sicher, dass Ihr Benutzer über die erforderlichen Berechtigungen verfügt, um Daten aus Ihrem HDFS-Quelldienst zu migrieren. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Hive als Quelle.

Das folgende Codebeispiel erstellt einen Tunnel für die AWS SCT Arbeit mit Ihrem Apache HDFS-Dienst. In diesem Beispiel wird der HADOOP_SOURCE Quellcluster verwendet, den Sie zuvor erstellt haben.

```
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'
/
```

Der folgende Code stellt ohne Tunnel eine Verbindung zu Ihrem Apache HDFS-Dienst her.

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

Ersetzen Sie in den vorherigen Beispielen *hdfs_user* und *hdfs_password* durch den Namen Ihres HDFS-Benutzers und das Passwort für diesen Benutzer.

Ersetzen Sie als Nächstes *hdfs_address* und *hdfs_port* durch die NameNode IP-Adresse und den Port Ihres Quell-Hadoop-Clusters.

Für *hdfs_remote_address* können Sie den Standardwert `127.0.0.1` oder die NameNode IP-Adresse Ihres Quell-Hive-Dienstes verwenden.

4. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihren Amazon EMR-Zielcluster und die Migrationsbefehle hinzu.

Verbindung zu Amazon EMR als Ziel herstellen

Sie können mit der AWS SCT CLI eine Verbindung zu Ihrem Amazon EMR-Zielcluster herstellen. Dazu autorisieren Sie eingehenden Datenverkehr und verwenden SSH. In diesem Fall AWS SCT verfügt es über alle erforderlichen Berechtigungen, um mit Ihrem Amazon EMR-Cluster zu arbeiten. Weitere Informationen finden Sie unter Bevor Sie Connect SSH mit dem primären Knoten verbinden und eine Verbindung zum primären Knoten herstellen im Amazon EMR Management Guide.

AWS SCT stellt mit dem hadoop Cluster-Benutzer eine Verbindung zu Amazon EMR Hive her. Um eine Verbindung zu Amazon EMR Hive herzustellen, verwenden Sie den Hive-JDBC-Treiber Version 2.6.2.1002 oder höher. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

So stellen Sie in der CLI eine Verbindung zu Amazon EMR her AWS SCT

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihren Hadoop-Quellcluster enthält. Fügen Sie die Amazon EMR-Zielanmeldedaten zu dieser Datei hinzu.
2. Fügen Sie dem Projekt Ihren Amazon EMR-Zielcluster hinzu.

   Das folgende Codebeispiel fügt den Amazon EMR-Zielcluster hinzu. In diesem HADOOP_TARGET Beispiel wird der Name des Zielclusters festgelegt. Verwenden Sie diesen Objektnamen, um Ihre Hive- und HDFS-Services sowie einen Amazon S3 S3-Bucket-Ordner zum Projekt hinzuzufügen und Zuordnungsregeln zu erstellen.

```
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'
/
```

Geben Sie im vorherigen Beispiel Ihre AWS Ressourcennamen und Amazon EMR-Verbindungsinformationen ein. Dazu gehören die IP-Adresse Ihres Amazon EMR-Clusters, der AWS Zugriffsschlüssel, der AWS geheime Zugriffsschlüssel und der Amazon S3 S3-Bucket. Konfigurieren Sie bei Bedarf den Wert der Port-Variablen. Ersetzen Sie anschließend *emr_user* und *emr_password* durch den Namen Ihres Amazon EMR-Benutzers und das Passwort für diesen Benutzer. Geben Sie für *path\name* den Namen und den Pfad zur PEM-Datei für Ihren Amazon EMR-Zielcluster ein. Weitere Informationen finden Sie unter PEM-Datei für EMR-Clusterzugriff herunterladen.

3. Fügen Sie dem Projekt Ihren Amazon S3 S3-Ziel-Bucket hinzu.

   Das folgende Codebeispiel fügt den Amazon S3 S3-Ziel-Bucket hinzu. In diesem Beispiel wird der HADOOP_TARGET Cluster verwendet, den Sie zuvor erstellt haben.

   ```
   AddTargetClusterS3
    -cluster: 'HADOOP_TARGET'
    -Name: 'S3_TARGET'
    -accessKey: 'AKIAIOSFODNN7EXAMPLE'
    -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY'
    -region: 'eu-west-1'
    -s3Path: 'doc-example-bucket/example-folder'
   /
   ```

   Geben Sie im vorherigen Beispiel Ihren Zugriffsschlüssel, Ihren AWS geheimen AWS Zugriffsschlüssel und Ihren Amazon S3 S3-Bucket ein.

4. Fügen Sie dem Projekt Ihren Hive-Zieldienst hinzu.

   Das folgende Codebeispiel erstellt einen Tunnel, AWS SCT um mit Ihrem Hive-Zieldienst zu arbeiten. In diesem Beispiel wird der HADOOP_TARGET Zielcluster verwendet, den Sie zuvor erstellt haben.

   ```
   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'
/
```

Ersetzen Sie im vorherigen Beispiel *hive_user* und *hive_password* durch den Namen Ihres Hive-Benutzers und das Passwort für diesen Benutzer.

Als Nächstes *hive_address* ersetzen Sie es durch den Standardwert `127.0.0.1` oder durch die NameNode IP-Adresse Ihres Hive-Zieldienstes. Als Nächstes *hive_port* ersetzen Sie ihn durch den Port Ihres Hive-Zieldienstes.

5. Fügen Sie dem Projekt Ihren Ziel-HDFS-Dienst hinzu.

   Das folgende Codebeispiel erstellt einen Tunnel für die AWS SCT Arbeit mit Ihrem Apache HDFS-Dienst. In diesem Beispiel wird der `HADOOP_TARGET` Zielcluster verwendet, den Sie zuvor erstellt haben.

   ```
   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'
   /
   ```

   Ersetzen Sie im vorherigen Beispiel *hdfs_user* und *hdfs_password* durch den Namen Ihres HDFS-Benutzers und das Kennwort für diesen Benutzer.

   Ersetzen Sie als Nächstes *hdfs_address* und *hdfs_port* durch die private IP-Adresse und den Port NameNode Ihres Ziel-HDFS-Dienstes.

6. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes Zuordnungsregeln und Migrationsbefehle hinzu. Weitere Informationen finden Sie unter Migration von Hadoop-Workloads.

Herstellen einer Verbindung zu Apache Oozie-Workflows mit dem AWS Schema Conversion Tool

Sie können die AWS SCT Befehlszeilenschnittstelle (CLI) verwenden, um Apache Oozie-Workflows in zu AWS Step Functions konvertieren. Nachdem Sie Ihre Apache Hadoop-Workloads zu Amazon EMR migriert haben, können Sie einen nativen Service in der verwenden, um Ihre Jobs AWS Cloud zu orchestrieren. Weitere Informationen finden Sie unter Verbindung zu Apache Hadoop herstellen.

AWS SCT konvertiert Ihre Oozie-Workflows in Funktionen, die nicht unterstützt werden, AWS Step Functions und verwendet diese AWS Lambda , um sie zu emulieren. AWS Step Functions AWS SCT Konvertiert außerdem Ihre Oozie-Jobeigenschaften in. AWS Systems Manager

Um Apache Oozie-Workflows zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.671 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle von vertraut. AWS SCT Weitere Informationen finden Sie unter CLI-Referenz für AWS Schema Conversion Tool.

Voraussetzungen für die Verwendung von Apache Oozie als Quelle

Die folgenden Voraussetzungen sind erforderlich, um über die AWS SCT CLI eine Verbindung zu Apache Oozie herzustellen.

- Erstellen Sie einen Amazon S3 S3-Bucket, um die Definitionen von Zustandsmaschinen zu speichern. Sie können diese Definitionen verwenden, um Ihre Zustandsmaschinen zu konfigurieren. Weitere Informationen finden Sie unter Erstellen eines Buckets im Amazon S3 S3-Benutzerhandbuch.
- Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle mit der `AmazonS3FullAccess` Richtlinie. AWS SCT verwendet diese IAM-Rolle, um auf Ihren Amazon S3 S3-Bucket zuzugreifen.
- Notieren Sie sich Ihren AWS geheimen Schlüssel und Ihren AWS geheimen Zugriffsschlüssel. Weitere Informationen zu AWS Zugriffsschlüsseln finden Sie unter Verwaltung von Zugriffsschlüsseln im IAM-Benutzerhandbuch.
- Speichern Sie Ihre AWS Anmeldeinformationen und die Informationen zu Ihrem Amazon S3 S3-Bucket im AWS Serviceprofil in den globalen Anwendungseinstellungen. AWS SCT Verwendet dann dieses AWS Serviceprofil, um mit Ihren AWS Ressourcen zu arbeiten. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.

Um mit Ihren Apache Oozie-Quell-Workflows arbeiten zu können, AWS SCT ist die spezifische Struktur Ihrer Quelldateien erforderlich. Jeder Ihrer Anwendungsordner muss die `job.properties` Datei enthalten. Diese Datei enthält Schlüssel-Wert-Paare Ihrer Jobeigenschaften. Außerdem muss jeder Ihrer Bewerbungsordner die `workflow.xml` Datei enthalten. In dieser Datei werden die Aktionsknoten und Kontrollflussknoten Ihres Workflows beschrieben.

Verbindung zu Apache Oozie als Quelle herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihren Apache Oozie-Quelldateien herzustellen.

So stellen Sie in der CLI eine Verbindung zu Apache Oozie her AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können die `OozieConversionTemplate.scts` Vorlage beispielsweise herunterladen und bearbeiten. Weitere Informationen finden Sie unter CLI-Szenarien abrufen.
2. Konfigurieren Sie die AWS SCT Anwendungseinstellungen.

   Das folgende Codebeispiel speichert die Anwendungseinstellungen und ermöglicht das Speichern von Passwörtern in Ihrem Projekt. Sie können diese gespeicherten Einstellungen in anderen Projekten verwenden.

   ```
   SetGlobalSettings
       -save: 'true'
       -settings: '{
           "store_password": "true"
       }'
   /
   ```
3. Erstellen Sie ein neues AWS SCT Projekt.

   Im folgenden Codebeispiel wird das `oozie` Projekt im `c:\sct` Ordner erstellt.

   ```
   CreateProject
       -name: 'oozie'
       -directory: 'c:\sct'
   /
   ```
4. Fügen Sie den Ordner mit Ihren Apache Oozie-Quelldateien mithilfe des `AddSource` Befehls zum Projekt hinzu. Stellen Sie sicher, dass Sie den `APACHE_OOZIE` Wert für den `vendor`

Parameter verwenden. Geben Sie außerdem Werte für die folgenden erforderlichen Parameter an: `name` und`mappingsFolder`.

Das folgende Codebeispiel fügt Apache Oozie als Quelle in Ihr AWS SCT Projekt ein. In diesem Beispiel wird ein Quellobjekt mit dem Namen `OOZIE` erstellt. Verwenden Sie diesen Objektnamen, um Zuordnungsregeln hinzuzufügen. AWS SCT Verwendet nach der Ausführung dieses Codebeispiels den `c:\oozie` Ordner, um Ihre Quelldateien in das Projekt zu laden.

```
AddSource
    -name: 'OOZIE'
    -vendor: 'APACHE_OOZIE'
    -mappingsFolder: 'c:\oozie'
/
```

Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden.

5. Stellen Sie mit dem `ConnectSource` Befehl eine Connect zu Ihren Apache Oozie-Quelldateien her. Verwenden Sie den Namen Ihres Quellobjekts, den Sie im vorherigen Schritt definiert haben.

```
ConnectSource
    -name: 'OOZIE'
    -mappingsFolder: 'c:\oozie'
/
```

6. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihren AWS Step Functions Dienst hinzu.

Berechtigungen für die Verwendung von AWS Lambda Funktionen im Erweiterungspaket

Für die Quellfunktionen, die AWS Step Functions nicht unterstützt werden, AWS SCT erstellt ein Erweiterungspaket. Dieses Erweiterungspaket enthält AWS Lambda Funktionen, die Ihre Quellfunktionen emulieren.

Um dieses Erweiterungspaket zu verwenden, erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle mit den folgenden Berechtigungen.

```
{
    "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:::*/*"
            ]
        }
    ]
}
```

Um das Erweiterungspaket anzuwenden, AWS SCT ist eine IAM-Rolle mit den folgenden Berechtigungen erforderlich.

```
{
    "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"
            ]
        }
    ]
```

```
}
```

AWS Step Functions Als Ziel wird eine Verbindung hergestellt

Gehen Sie wie folgt vor, um eine Verbindung zu AWS Step Functions als Ziel herzustellen.

Um AWS Step Functions in der AWS SCT CLI eine Verbindung herzustellen

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihre Apache Oozie-Quelldateien enthält.
2. Fügen Sie dem AWS SCT Projekt mithilfe des `AddTarget` Befehls die Informationen zu Ihrem Migrationsziel hinzu. Stellen Sie sicher, dass Sie den `STEP_FUNCTIONS` Wert für den `vendor` Parameter verwenden. Geben Sie außerdem Werte für die folgenden erforderlichen Parameter an: `name` und`profile`.

   Das folgende Codebeispiel fügt AWS Step Functions Ihrem AWS SCT Projekt eine Quelle hinzu. In diesem Beispiel wird ein Zielobjekt mit dem Namen erstellt`AWS_STEP_FUNCTIONS`. Verwenden Sie diesen Objektnamen, wenn Sie Zuordnungsregeln erstellen. In diesem Beispiel wird außerdem ein AWS SCT Dienstprofil verwendet, das Sie im Schritt „Voraussetzungen“ erstellt haben. Stellen Sie sicher, dass Sie es *`profile_name`* durch den Namen Ihres Profils ersetzen.

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

   Wenn Sie das AWS Dienstprofil nicht verwenden, stellen Sie sicher, dass Sie Werte für die folgenden erforderlichen Parameter angeben: `accessKey`,`secretKey`,`awsRegion`, und`s3Path`. Verwenden Sie diese Parameter, um Ihren AWS geheimen Zugriffsschlüssel, Ihren AWS geheimen Schlüssel und den Pfad zu Ihrem Amazon S3 S3-Bucket anzugeben. AWS-Region
3. Stellen Sie AWS Step Functions mit dem `ConnectTarget` Befehl eine Connect. Verwenden Sie den Namen Ihres Zielobjekts, den Sie im vorherigen Schritt definiert haben.

Das folgende Codebeispiel stellt mithilfe Ihres AWS Dienstprofils eine Verbindung zum AWS_STEP_FUNCTIONS Zielobjekt her. Stellen Sie sicher, dass Sie es *profile_name* durch den Namen Ihres Profils ersetzen.

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

4. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes Zuordnungsregeln und Migrationsbefehle hinzu. Weitere Informationen finden Sie unter Konvertierung von Oozie-Workflows;.

Herstellen einer Verbindung zu Microsoft Azure SQL-Datenbanken mit dem AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode aus der Azure SQL-Datenbank in die folgenden Ziele zu konvertieren:

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

Themen

- Berechtigungen für Azure SQL Database als Quelle
- Verbindung zur Azure SQL-Datenbank als Quelle herstellen

Berechtigungen für Azure SQL Database als Quelle

Die für Azure SQL Database als Quelle erforderlichen Rechte lauten wie folgt:

- VIEW DEFINITION
- VIEW DATABASE STATE

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Die für MySQL- und PostgreSQL-Zieldatenbanken erforderlichen Rechte werden in den folgenden Abschnitten beschrieben.

- Rechte für MySQL als Zieldatenbank
- Rechte für PostgreSQL als Zieldatenbank

Verbindung zur Azure SQL-Datenbank als Quelle herstellen

Verwenden Sie das folgende Verfahren, um eine Verbindung zu Ihrer Azure SQL Database-Quelldatenbank mit dem herzustellen AWS Schema Conversion Tool.

Um eine Verbindung zu einer Azure SQL Database-Quelldatenbank herzustellen

1. Wählen Sie in der die AWS Schema Conversion Tool Option Quelle hinzufügen aus.
2. Wählen Sie Azure SQL Database und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen der Azure SQL Database-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Datenbank	Geben Sie den Datenbanknamen ein, zu dem Sie eine Verbindung herstellen möchten.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Herstellen einer Verbindung zu IBM DB2 for z/OS-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT damit Schemas, Codeobjekte und Anwendungscode von IBM Db2 for z/OS in die folgenden Ziele konvertieren.

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

Voraussetzungen für Db2 for z/OS als Quelldatenbank

Die Datenbankversion IBM Db2 for z/OS Version 12 Function Level 100 unterstützt die meisten neuen Funktionen von IBM Db2 for z/OS Version 12 nicht. Diese Datenbankversion bietet Unterstützung für den Fallback auf Db2 Version 11 und die gemeinsame Nutzung von Daten mit Db2 Version 11. Um die Konvertierung nicht unterstützter Funktionen von Db2 Version 11 zu vermeiden, empfehlen wir, dass Sie eine IBM Db2 for z/OS-Datenbankfunktion der Stufe 500 oder höher als Quelle für verwenden. AWS SCT

Sie können das folgende Codebeispiel verwenden, um die Version Ihrer IBM Db2 for z/OS-Quelldatenbank zu überprüfen.

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

Stellen Sie sicher, dass dieser Code Version `DSN12015` oder höher zurückgibt.

Sie können das folgende Codebeispiel verwenden, um den Wert des `APPLICATION COMPATIBILITY` speziellen Registers in Ihrer IBM Db2 for z/OS-Quelldatenbank zu überprüfen.

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

Stellen Sie sicher, dass dieser Code Version `V12R1M500` oder höher zurückgibt.

Rechte für Db2 for z/OS als Quelldatenbank

Die folgenden Rechte sind erforderlich, um eine Verbindung zu einer Db2 for z/OS-Datenbank herzustellen und Systemkataloge und Tabellen zu lesen:

- WÄHLEN SIE AUF SYSIBM.LOCATIONS
- WÄHLEN SIE AUF SYSIBM.SYSCHECKS
- WÄHLEN SIE AUF SYSIBM.SYSCOLUMNS
- WÄHLEN SIE AUF SYSIBM.SYSDATABASE
- WÄHLEN SIE AUF SYSIBM.SYSDATATYPES
- WÄHLEN SIE AUF SYSIBM. SYSDUMMY1
- WÄHLEN SIE AUF SYSIBM.SYSFOREIGNKEYS
- WÄHLEN SIE AUF SYSIBM.SYSINDEXES
- WÄHLEN SIE AUF SYSIBM.SYSKEYCOLUSE
- WÄHLEN SIE AUF SYSIBM.SYSKEYS
- WÄHLEN SIE AUF SYSIBM.SYSKEYTARGETS
- WÄHLEN SIE AUF SYSIBM.SYSJAROBJECTS
- WÄHLEN SIE AUF SYSIBM.SYSPACKAGE
- WÄHLEN SIE AUF SYSIBM.SYSPARMS
- WÄHLEN SIE AUF SYSIBM.SYSRELS
- WÄHLEN SIE AUF SYSIBM.SYSROUTINES
- WÄHLEN SIE AUF SYSIBM.SYSSEQUENCES
- WÄHLEN SIE AUF SYSIBM.SYSSEQUENCESDEP
- WÄHLEN SIE AUF SYSIBM.SYSSYNONYMS
- WÄHLEN SIE AUF SYSIBM.SYSTABCONST
- WÄHLEN SIE AUF SYSIBM.SYSTABLES
- WÄHLEN SIE AUF SYSIBM.SYSTABLESPACE
- WÄHLEN SIE AUF SYSIBM.SYSTRIGGERS
- WÄHLEN SIE AUF SYSIBM.SYSVARIABLES
- WÄHLEN SIE AUF SYSIBM.SYSVIEWS

Um Db2 for z/OS-Tabellen in partitionierte PostgreSQL-Tabellen zu konvertieren, sammeln Sie mit dem Hilfsprogramm Statistiken über Tablespaces und Tabellen in Ihrer Datenbank, wie im Folgenden gezeigt. `RUNSTATS`

```
LISTDEF YOURLIST INCLUDE TABLESPACES DATABASE YOURDB
RUNSTATS TABLESPACE
```

```
LIST YOURLIST
TABLE (ALL) INDEX (ALL KEYCARD)
UPDATE ALL
REPORT YES
SHRLEVEL REFERENCE
```

Ersetzen Sie im vorherigen Beispiel den *YOURDB* Platzhalter durch den Namen der Quelldatenbank.

Verbindung zu Db2 for z/OS als Quelle herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihrer Db2 for z/OS-Quelldatenbank herzustellen. AWS SCT

So stellen Sie eine Verbindung zu einer IBM Db2 for z/OS-Quelldatenbank her

1. Wählen Sie in der die Option AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Db2 for z/OS und dann Weiter.

   Das Dialogfeld Quelle hinzufügen wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur IBM Db2 for z/OS-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.

Parameter	Aktion
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Ort	Geben Sie den eindeutigen Namen des Db2-Standorts ein, auf den Sie zugreifen möchten.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgeford ert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustellen.
Use SSL	Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informati onen ein: • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Speicherort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen.

Parameter	Aktion
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Treiberpfad für Db2 für z/OS	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *

- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- SELECT ON mysql.proc
- EINFÜGEN, AUF AWS_DB 2ZOS_EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_DB 2ZOS_EXT_DATA. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_DB 2ZOS_EXT_DATA. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Um Amazon RDS for MySQL als Ziel zu verwenden, setzen Sie den `log_bin_trust_function_creators` Parameter auf true und `character_set_server` auf`latin1`. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Um Aurora MySQL als Ziel zu verwenden, setzen Sie den `log_bin_trust_function_creators` Parameter auf true und `character_set_server` auf`latin1`. Setzen Sie den

`lower_case_table_names` Parameter außerdem auf true. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um Amazon RDS for PostgreSQL als Ziel zu verwenden, ist die `rds_superuser` entsprechende AWS SCT Berechtigung erforderlich.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *`user_name`* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *`db_name`* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *`your_password`* durch ein sicheres Passwort.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von Db2 for z/OS nach PostgreSQL

Um die Einstellungen für die Konvertierung von Db2 for z/OS nach PostgreSQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen

Liste Db2 for z/OS und dann Db2 for z/OS — PostgreSQL oder Db2 for z/OS — Amazon Aurora (PostgreSQL-kompatibel) aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von IBM Db2 for z/OS nach PostgreSQL an.

Die Einstellungen für die Konvertierung von Db2 for z/OS nach PostgreSQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um eindeutige Namen für Einschränkungen in der Zieldatenbank zu generieren.

  In PostgreSQL müssen alle von Ihnen verwendeten Einschränkungsnamen eindeutig sein. AWS SCT kann eindeutige Namen für Einschränkungen im konvertierten Code generieren, indem dem Namen Ihrer Einschränkung ein Präfix mit dem Tabellennamen hinzugefügt wird. Um sicherzustellen, dass dadurch eindeutige Namen für Ihre Einschränkungen AWS SCT generiert werden, wählen Sie Eindeutige Namen für Einschränkungen generieren aus.
- Um die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen im konvertierten Code beizubehalten.

  AWS SCT kann das Layout von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen an derselben Position und Reihenfolge wie im Quellcode beibehalten. Wählen Sie dazu für Die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen beibehalten die Option Ja aus.
- Um Tabellenpartitionen aus dem Konvertierungsbereich auszuschließen.

  AWS SCT kann während der Konvertierung alle Partitionen einer Quelltabelle überspringen. Wählen Sie dazu Tabellenpartitionen aus dem Konvertierungsbereich ausschließen aus.
- Um die automatische Partitionierung für Tabellen zu verwenden, die nach Wachstum partitioniert sind.

AWS SCT Kann für die Datenmigration automatisch alle Tabellen partitionieren, die größer als die angegebene Größe sind. Um diese Option zu verwenden, wählen Sie Partition von Tabellen erzwingen, die größer als sind, und geben Sie die Tabellengröße in Gigabyte ein. Geben Sie als Nächstes die Anzahl der Partitionen ein. AWS SCT berücksichtigt die Größe des Direct Access Storage Device (DASD) Ihrer Quelldatenbank, wenn Sie diese Option aktivieren.

AWS SCT kann die Anzahl der Partitionen automatisch ermitteln. Wählen Sie dazu die Option Anzahl der Partitionen proportional erhöhen und geben Sie die maximale Anzahl von Partitionen ein.

- Um dynamische Ergebnismengen als Array von Werten des Refcursor-Datentyps zurückzugeben.

  AWS SCT kann Quellprozeduren, die dynamische Ergebnismengen zurückgeben, in Prozeduren konvertieren, die ein Array von offenen Recursorn als zusätzlichen Ausgabeparameter haben. Wählen Sie dazu ein Array von Recursorn verwenden, um alle dynamischen Ergebnismengen zurückzugeben.

- Um den Standard anzugeben, der für die Konvertierung von Datums- und Uhrzeitwerten in Zeichenkettendarstellungen verwendet werden soll.

  AWS SCT kann Datums- und Uhrzeitwerte mithilfe eines der unterstützten Branchenformate in Zeichenfolgendarstellungen konvertieren. Wählen Sie dazu Zeichenfolgendarstellungen von Datumswerten verwenden oder Zeichenfolgendarstellungen von Zeitwerten verwenden aus. Wählen Sie als Nächstes einen der folgenden Standards aus.

  - Internationale Organisation für Normung (ISO)
  - IBM Europäischer Standard (EUR)
  - IBM USA Standard (Vereinigte Staaten)
  - Japanischer Industriestandard Christian Era (JIS)

ConnConnecting zu IBM DB2 für Linux-, UNIX- und Windows-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte in der SQL-Sprache und Anwendungscode von IBM Db2 für Linux, Unix und Windows (Db2 LUW) in die folgenden Ziele zu konvertieren.

- Amazon RDS für MySQL

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

AWS SCT unterstützt als Quelle die Db2-LUW-Versionen 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 und 11.5.

Rechte für Db2 LUW als Quelle

Die folgenden Rechte sind erforderlich, um eine Verbindung zu einer Db2 LUW-Datenbank herzustellen, die verfügbaren Rechte zu überprüfen und Schema-Metadaten für eine Quelle zu lesen:

- Berechtigung, die erforderlich sind, um eine Verbindung herzustellen:
  - VERBINDUNG ZUR DATENBANK HERSTELLEN
- Berechtigung, die zum Ausführen von SQL-Anweisungen erforderlich ist:
  - AUF PAKET NULLID AUSFÜHREN. SYSSH200
- Berechtigungen, die für den Abruf von Informationen auf Instance-Ebene erforderlich sind:
  - FÜHRE DIE FUNKTION SYSPROC.ENV_GET_INST_INFO AUS
  - WÄHLEN SIE AUF SYSIBMADM.ENV_INST_INFO
  - WÄHLEN SIE AUF SYSIBMADM.ENV_SYS_INFO
- Berechtigungen, die benötigt werden, um Berechtigungen zu prüfen, die durch Rollen, Gruppen und Behörden vergeben werden:
  - FÜHRE DIE FUNKTION SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID AUS
  - FÜR DIE FUNKTION SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID AUSFÜHREN
  - FÜR DIE FUNKTION SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID AUSFÜHREN
  - WÄHLEN SIE AUF SYSIBMADM.PRIVILEGES
- Berechtigungen, die für System-Kataloge und Tabellen benötigt werden:
  - WÄHLEN SIE AUF SYSCAT.ATTRIBUTES
  - WÄHLEN SIE AUF SYSCAT.CHECKS
  - WÄHLEN SIE AUF SYSCAT.COLIDENTATTRIBUTES
  - WÄHLEN SIE AUF SYSCAT.COLUMNS
  - WÄHLEN SIE FÜR SYSCAT.DATAPARTITIONEXPRESSION

- WÄHLEN SIE AUF SYSCAT.DATAPARTITIONS
- WÄHLEN SIE AUF SYSCAT.DATATYPEDEP
- WÄHLEN SIE SYSCAT.DATATYPES
- WÄHLEN SIE AUF SYSCAT.HIERARCHIES
- WÄHLEN SIE AUF SYSCAT.INDEXCOLUSE
- WÄHLEN SIE AUF SYSCAT.INDEXES
- WÄHLEN SIE AUF SYSCAT.INDEX-PARTITIONEN
- WÄHLEN SIE AUF SYSCAT.KEYCLUSE
- WÄHLEN SIE AUF SYSCAT.MODULEOBJECTS
- WÄHLEN SIE AUF SYSCAT.MODULES
- WÄHLEN SIE AUF SYSCAT.NICKNAMES
- WÄHLEN SIE AUF SYSCAT.PERIODS
- WÄHLEN SIE AUF SYSCAT.REFERENCES
- WÄHLEN SIE AUF SYSCAT.ROUTINEPARMS
- WÄHLEN SIE AUF SYSCAT.ROUTINES
- WÄHLEN SIE AUF SYSCAT.ROWFIELDS
- WÄHLEN SIE AUF SYSCAT.SCHEMATA
- WÄHLEN SIE AUF SYSCAT.SEQUENCES
- WÄHLEN SIE AUF SYSCAT.TABCONST
- WÄHLEN SIE AUF SYSCAT.TABLES
- WÄHLEN SIE AUF SYSCAT.TRIGGERS
- WÄHLEN SIE AUF SYSCAT.VARIABLEDEP
- WÄHLEN SIE AUF SYSCAT.VARIABLES
- WÄHLEN SIE AUF SYSCAT.VIEWS
- WÄHLEN SIE AUF SYSIBM. SYSDUMMY1

- Für die Ausführung von SQL-Anweisungen benötigt das Benutzerkonto eine Berechtigung, mindestens eine der in der Datenbank aktivierten Arbeitslasten zu verwenden. Wurde dem Benutzer keine der Arbeitslasten zugeordnet, stellen Sie sicher, dass der Benutzer Zugriff auf die Standard-Benutzerarbeitslast hat:
  - VERWENDUNG AUF DER ARBEITSLAST SYSDEFAULTUSERWORKLOAD

Um Abfragen auszuführen, müssen Sie temporäre System-Tabellenräume mit den Seitengrößen 8K, 16K und 32K anlegen, falls diese nicht existieren. Um die temporären Tabellenräume zu erstellen, führen Sie die folgenden Skripts aus:

```
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
  PAGESIZE 32768
  BUFFERPOOL BP32K;
```

Verbindung zu Db2 LUW als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Db2 LUW-Quelldatenbank herzustellen.

Herstellen einer Verbindung zu einer Db2 LUW-Quelldatenbank

1. Wählen Sie in der die Option AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Db2 LUW und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur IBM Db2 LUW-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Db2 LUW-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass

Parameter	Aktion
	das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgeford ert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustellen.
Use SSL	Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informati onen ein: • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Speicherort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Db2-LUW-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.

6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Migration von IBM DB2 für Linux, UNIX und Windows zu Amazon Relational Database Service for PostgreSQL oder Amazon Aurora PostgreSQL-Compatible Edition

Wenn Sie IBM Db2 LUW zu PostgreSQL migrieren, AWS SCT können Sie verschiedene Trigger-Anweisungen konvertieren, die mit Db2 LUW verwendet werden. Zu diesen Auslöseranweisungen gehören:

- Triggerereignisse — Die Triggerereignisse INSERT, DELETE und UPDATE geben an, dass die ausgelöste Aktion immer dann ausgeführt wird, wenn das Ereignis auf die Betrefftabelle oder Betreffansicht angewendet wird. Sie können eine beliebige Kombination der INSERT-, DELETE- und UPDATE-Ereignisse angeben, aber Sie können jedes Ereignis nur einmal angeben. AWS SCT unterstützt einzelne und mehrere Triggerereignisse. Bei Ereignissen hat PostgreSQL praktisch die gleiche Funktionalität.
- Event OF COLUMN — Sie können einen Spaltennamen aus einer Basistabelle angeben. Der Auslöser wird nur durch die Aktualisierung einer Spalte ausgelöst, die in der Liste der Spaltennamen identifiziert wird. PostgreSQL hat dieselbe Funktionalität.
- Trigger für Anweisungen — Diese geben an, dass die ausgelöste Aktion nur einmal für die gesamte Anweisung angewendet wird. Sie können diese Art von Auslösergranularität nicht für einen BEFORE-Auslöser oder einen INSTEAD OF-Auslöser anwenden. Sofern dieser Parameter angegeben ist, wird ein UPDATE- oder DELETE-Auslöser aktiviert. Dies gilt auch dann, wenn keine Zeilen betroffen sind. PostgreSQL verfügt auch über diese Funktionalität und die Auslöserdeklaration für Anweisungsauslöser ist für PostgreSQL und Db2 LUW identisch.
- Referenzierungsklauseln — Diese geben die Korrelationsnamen für Übergangsvariablen und die Tabellennamen für Übergangstabellen an. Korrelationsnamen identifizieren eine bestimmte Zeile in der Gruppe von Zeilen, die von der auslösenden SQL-Operation betroffen sind. Tabellennamen identifizieren die vollständige Menge der betroffenen Zeilen. Jede durch eine auslösende SQL-Operation betroffene Zeile ist für die ausgelöste Aktion anhand qualifizierender Spalten mit angegebenen Korrelationsnamen verfügbar. PostgreSQL unterstützt diese Funktionalität nicht und verwendet nur den Korrelationsnamen NEW oder OLD.
- INSTEAD OF-Trigger — AWS SCT unterstützt diese.

Konvertierung von partitionierten Db2-LUW-Tabellen in partitionierte Tabellen nach PostgreSQL Version 10

AWS SCT kann Db2-LUW-Tabellen in partitionierte Tabellen in PostgreSQL 10 konvertieren. Es gibt mehrere Einschränkungen beim Umwandeln einer partitionierten Db2 LUW-Tabelle in PostgreSQL:

- Sie können in Db2 LUW eine partitionierte Tabelle mit einer löschbaren Spalte erstellen, und Sie können eine Partition angeben, die NULL-Werte speichert. PostgreSQL unterstützt jedoch keine NULL-Werte für RANGE-Partitionierungen.
- LUW Db2 kann in einer INCLUSIVE- oder EXCLUSIVE-Klausel Bereichsgrenzenwerte festlegen. PostgreSQL unterstützt nur INCLUSIVE für ein Startgrenze und EXCLUSIVE für eine Endgrenze. Der Name der konvertierten Partition hat das Format <original_table_name>_<original_partition_name>.
- Für partitionierte Tabellen in Db2 LUW können Sie primäre oder eindeutige Schlüssel erstellen. In PostgreSQL müssen Sie primäre oder eindeutige Schlüssel direkt für jede Partition erstellen. Primäre oder eindeutige Schlüsselbeschränkungen müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Schlüssels hat das Format <original_key_name>_<original_partition _name>.
- Sie können eine Fremdschlüsselbeschränkung von und zu einer partitionierten Tabelle in der Db2-LUW erstellen. PostgreSQL unterstützt jedoch keine Fremdschlüsselverweise in partitionierten Tabellen. PostgreSQL unterstützt auch keine Fremdschlüsselverweise aus einer partitionierten Tabelle in eine andere Tabelle.
- Sie können einen Index für eine partitionierte Tabelle in Db2 LUW erstellen. In PostgreSQL müssen Sie jedoch einen Index direkt für jede Partition erstellen. Indizes müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Index hat das Format <original_index_name>_<original_partition_name>.
- Sie müssen Zeilenauslöser für einzelne Partitionen definieren, nicht für die partitionierte Tabelle. Auslöser müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Auslösers hat das Format <original_trigger_name>_<original_partition_name>.

Berechtigungen für PostgreSQL als Ziel

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *`user_name`* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *`db_name`* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *`your_password`* durch ein sicheres Passwort.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Migration von IBM DB2 für Linux, UNIX und Windows zu Amazon RDS for MySQL oder Amazon Aurora MySQL

Beachten Sie Folgendes, wenn Sie eine IBM Db2 LUW-Datenbank in RDS for MySQL oder Amazon Aurora MySQL konvertieren.

Berechtigungen für MySQL als Ziel

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *
- SELECT ON *.*

- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- SELECT ON mysql.proc
- EINFÜGEN, AUF AWS_DB 2_EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_DB 2_EXT_DATA. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_DB 2_EXT_DATA. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Setzen Sie den Parameter `lower_case_table_names` auf 1, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen

von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf 1. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Verwendung von MySQL als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode von MySQL in die folgenden Ziele zu konvertieren:

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

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- Rechte für MySQL als Quelldatenbank
- Verbindung zu MySQL als Quelle herstellen
- Rechte für PostgreSQL als Zieldatenbank

Rechte für MySQL als Quelldatenbank

Die für MySQL als Quelle erforderlichen Rechte lauten wie folgt:

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

Verbindung zu MySQL als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer MySQL-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer MySQL-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen.
2. Wählen Sie MySQL und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen für die MySQL-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein. Sie können über ein IPv6 Adressprotokoll eine Verbindung zu Ihrer MySQL-Quelldatenbank herstellen. Stellen Sie dazu sicher, dass Sie bei der Eingabe der IP-Adresse eckige Klammern verwenden, wie im folgenden Beispiel gezeigt. `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.

Parameter	Aktion
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgeford ert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustellen.
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • SSL erforderlich: Wählen Sie diese Option, um nur über SSL eine Verbindung zum Server herzustellen. Wenn Sie Require SSL aktivieren, ist eine Verbindung zum Server nur möglich, wenn der Server SSL unterstüt zt. Wenn Sie Require SSL nicht aktivieren und der Server SSL nicht unterstützt, können Sie sich ohne SSL mit dem Server verbinden. Weitere Informationen finden Sie unter Konfiguration von MySQL für die Verwendung sicherer Verbindungen. • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Vertrauensspeichers zu überprüfen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält.

Parameter	Aktion
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank passwort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.
MySql Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *db_name* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *your_password* durch ein sicheres Passwort.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Herstellen einer Verbindung zu Oracle-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode aus der Oracle-Datenbank in die folgenden Ziele zu konvertieren:

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

Wenn die Quelle eine Oracle-Datenbank ist, können Kommentare in das entsprechende Format konvertiert werden, z. B. in einer PostgreSQL-Datenbank. AWS SCT kann Kommentare zu Tabellen, Ansichten und Spalten konvertieren. Kommentare können Apostrophe enthalten. Bei der Konvertierung von SQL-Anweisungen werden die Apostrophe AWS SCT verdoppelt, genau wie bei Zeichenkettenliteralen.

Weitere Informationen finden Sie unter den folgenden Topics.

Themen

- Berechtigungen für Oracle als Quelle

- Verbindung zu Oracle als Quelle herstellen
- Migration von Oracle zu Amazon RDS for PostgreSQL oder Amazon Aurora PostgreSQL mit AWS Schema Conversion Tool
- Migration von Oracle zu Amazon RDS for MySQL oder Amazon Aurora MySQL mit dem AWS Schema Conversion Tool
- Migration von Oracle Database zu Amazon RDS for Oracle mit AWS Schema Conversion Tool

Berechtigungen für Oracle als Quelle

Für Oracle als Quelle sind folgende Rechte erforderlich:

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT$

Verbindung zu Oracle als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Oracle-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Oracle-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen aus.
2. Wählen Sie Oracle und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Oracle-Quelldatenbank manuell einzugeben:

Parameter	Aktion

Typ	Wählen Sie den Verbindungstyp für die Datenbank aus. Geben Sie abhängig vom gewählten Typ die folgenden zusätzlichen Informationen ein:
	• SID
	  • Servername: Der DNS-Name (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers.
	  • Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird
	  • Oracle SID: Die Oracle-System-ID (SID). Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank:
	  `SELECT sys_context('userenv','instance_name') AS SID FROM dual;`
	• Name des Dienstes
	  • Server name: Der DNS-Name oder die IP-Adresse Ihres Quelldatenbankservers
	  Sie können mithilfe eines IPv6 Adressprotokolls eine Verbindung zu Ihrer Oracle-Quelldatenbank herstellen. Stellen Sie dazu sicher, dass Sie bei der Eingabe der IP-Adresse eckige Klammern verwenden, wie im folgenden Beispiel gezeigt.
	  `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
	  • Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird
	  • Dienstname: Der Name des Oracle-Dienstes, zu dem eine Verbindung hergestellt werden soll.
	• TNS-Alias

Parameter	Aktion
	• TNS file path: Der Pfad zu der Datei, die die Transparent Network Substrate (TNS)-Namensverbindungsinformationen enthält Nachdem Sie die TNS-Datei ausgewählt haben, werden alle Oracle-Datenbankverbindungen aus der Datei zur TNS-Aliasliste AWS SCT hinzugefügt. Wählen Sie diese Option, um eine Verbindung zu Oracle Real Application Clusters (RAC) herzustellen. • TNS-Alias: Der TNS-Alias aus dieser Datei, der für die Verbindung mit der Quelldatenbank verwendet werden soll. • TNS-Connect-ID • TNS Connect-ID: Die Kennung für die registrierten TNS-Verbindungsinformationen.

Parameter	Aktion
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. Wenn Sie sich zum ersten Mal mit der Oracle-Datenbank verbinden, geben Sie den Pfad zur Oracle-Treiberdatei (ojdbc8.jar) ein. Sie können die Datei unter http://www.oracle herunterladen. com/technetwork/database/features/jdbc/index-091264.html. Stellen Sie sicher, dass Sie sich auf der kostenlosen Website von Oracle Technical Network registrieren, um den Download abzuschließen. AWS SCT verwendet den ausgewählten Treiber für alle future Oracle-Datenbankverbindungen. Der Treiberpfad kann mithilfe der Registerkarte Treiber in den globalen Einstellungen geändert werden. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgefordert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • SSL-Authentifizierung: Wählen Sie diese Option, um die SSL-Authentifizierung per Zertifikat zu verwenden . Richten Sie Ihren Vertrauensspeicher und Schlüssel speicher unter Einstellungen, Allgemeine Einstellungen, Sicherheit ein. • Trust Store: Der zu verwendende Trust Store. • Schlüsselspeicher: Der zu verwendende Schlüssel speicher.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wählen Sie diese Option, um das Datenbankkennwort zu speichern und schnell eine Verbindung zur Datenbank herzustellen, ohne das Passwort eingeben zu müssen.
Oracle-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Treiberpfad nicht im Verbindungsdialogfeld angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Migration von Oracle zu Amazon RDS for PostgreSQL oder Amazon Aurora PostgreSQL mit AWS Schema Conversion Tool

Beachten Sie Folgendes, wenn Sie eine Oracle-Datenbank in RDS for PostgreSQL oder Amazon Aurora PostgreSQL konvertieren.

Themen

- Rechte für PostgreSQL als Zieldatenbank
- Einstellungen für die Konvertierung von Oracle in PostgreSQL
- Oracle-Sequenzen konvertieren
- Konvertieren von Oracle ROWID
- Oracle Dynamic SQL wird konvertiert
- Oracle-Partitionen konvertieren

AWS SCT Führt bei der Konvertierung von Oracle-Systemobjekten nach PostgreSQL Konvertierungen durch, wie in der folgenden Tabelle dargestellt.

Oracle-Systemobjekt	Beschreibung	Konvertiertes PostgreSQL-Objekt
V$VERSION	Zeigt die Versionsnummern der Core-Bibliothekskomponenten in der Oracle-Datenbank an	aws_oracle_ext.v$version
V$INSTANCE	Eine Ansicht, die den Status der aktuellen Instance anzeigt.	aws_oracle_ext.v$instance

Sie können AWS SCT es verwenden, um Oracle SQL*Plus-Dateien in psql zu konvertieren, ein terminalbasiertes Frontend für PostgreSQL. Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren mit AWS SCT.

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *`user_name`* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *`db_name`* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *`your_password`* durch ein sicheres Passwort.

Um Amazon RDS for PostgreSQL als Ziel zu verwenden, ist die `rds_superuser` entsprechende AWS SCT Berechtigung erforderlich.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von Oracle in PostgreSQL

Um die Konvertierungseinstellungen von Oracle nach PostgreSQL zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle nach PostgreSQL an.

Die Konvertierungseinstellungen von Oracle nach PostgreSQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus.

AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Um die Konvertierung von materialisierten Oracle-Ansichten in Tabellen oder materialisierte Ansichten auf PostgreSQL zu ermöglichen AWS SCT . Wählen Sie für Materialized View Conversion as aus, wie Ihre materialisierten Quellansichten konvertiert werden sollen.
- Um mit Ihrem Oracle-Quellcode zu arbeiten, wenn er die `TO_NUMBER` Funktionen `TO_CHAR``TO_DATE`, und mit Parametern enthält, die PostgreSQL nicht unterstützt. AWS SCT Emuliert standardmäßig die Verwendung dieser Parameter im konvertierten Code.

  Wenn Ihr Oracle-Quellcode nur Parameter enthält, die PostgreSQL unterstützt, können Sie native PostgreSQL-`TO_CHAR`, `TO_DATE` und -Funktionen verwenden. `TO_NUMBER` In diesem Fall arbeitet der konvertierte Code schneller. Um nur diese Parameter einzubeziehen, wählen Sie die folgenden Werte aus:

  - Die Funktion TO_CHAR () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
  - Die Funktion TO_DATE () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
  - Die Funktion TO_NUMBER () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
- Um zu vermeiden, dass Ihre Oracle-Quelldatenbank nur Ganzzahlwerte in den Primär- oder Fremdschlüsselspalten des `NUMBER` Datentyps speichert, AWS SCT können Sie diese Spalten in den `BIGINT` Datentyp konvertieren. Dieser Ansatz verbessert die Leistung des konvertierten Codes. Wählen Sie für diesen Ansatz die Option NUMBER-Primär-/Fremdschlüsselspalten in BIGINT-Spalten konvertieren aus. Stellen Sie sicher, dass Ihre Quelle keine Gleitkommawerte in diesen Spalten enthält, um Datenverlust zu vermeiden.
- Um deaktivierte Trigger und Einschränkungen in Ihrem Quellcode zu überspringen. Wählen Sie dazu Deaktivierte Trigger und Einschränkungen ignorieren.
- Wird verwendet AWS SCT , um Zeichenkettenvariablen zu konvertieren, die als dynamisches SQL aufgerufen werden. Ihr Datenbankcode kann die Werte dieser Zeichenfolgenvariablen ändern. Um sicherzustellen, dass AWS SCT immer der neueste Wert dieser Zeichenkettenvariablen konvertiert wird, wählen Sie Dynamischen SQL-Code konvertieren, der in aufgerufenen Routinen erstellt wurde.
- Um dieses Problem zu beheben, unterstützen PostgreSQL Version 10 und früher keine Verfahren. Wenn Sie oder Ihre Benutzer mit der Verwendung von Prozeduren in PostgreSQL nicht vertraut

sind, AWS SCT können Sie Oracle-Prozeduren in PostgreSQL-Funktionen konvertieren. Wählen Sie dazu Verfahren in Funktionen konvertieren aus.

- Um zusätzliche Informationen zu den aufgetretenen Aktionspunkten zu erhalten. Zu diesem Zweck können Sie dem Erweiterungspaket bestimmte Funktionen hinzufügen, indem Sie bei Migrationsproblemen mit den nächsten Schweregraden die Option Hinzufügen bei Ausnahmeauslösung auswählen. Wählen Sie anschließend Schweregrade zum Auslösen benutzerdefinierter Ausnahmen aus.
- Um mit einer Oracle-Quelldatenbank zu arbeiten, die Einschränkungen mit den automatisch generierten Namen enthalten kann. Wenn Ihr Quellcode diese Namen verwendet, stellen Sie sicher, dass Sie die Option Vom System generierte Beschränkungsnamen mithilfe der ursprünglichen Quellnamen konvertieren auswählen. Wenn Ihr Quellcode diese Einschränkungen verwendet, aber nicht ihre Namen verwendet, deaktivieren Sie diese Option, um die Konvertierungsgeschwindigkeit zu erhöhen.
- Um zu überprüfen, ob Ihre Datenbank und Anwendungen in unterschiedlichen Zeitzonen ausgeführt werden. AWS SCT Emuliert standardmäßig Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. In diesem Fall entspricht die Option Zeitzone auf der Clientseite der Zeitzone auf dem Server.
- Um zu überprüfen, ob Ihre Quell- und Zieldatenbanken in unterschiedlichen Zeitzonen laufen. Wenn dies der Fall ist, gibt die Funktion, die die `SYSDATE` integrierte Oracle-Funktion emuliert, andere Werte zurück als die Quellfunktion. Um sicherzustellen, dass Ihre Quell- und Zielfunktionen dieselben Werte zurückgeben, wählen Sie Standardzeitzone für die SYSDATE-Emulation festlegen.
- Um die Funktionen der Oracle-Erweiterung in Ihrem konvertierten Code zu verwenden. Wählen Sie dazu unter Orace-Implementierung verwenden die zu verwendenden Funktionen aus. Weitere Informationen zu Oracle finden Sie unter orafce on. GitHub

Oracle-Sequenzen konvertieren

AWS SCT konvertiert Sequenzen von Oracle nach PostgreSQL. Wenn Sie Sequenzen verwenden, um Integritätseinschränkungen aufrechtzuerhalten, stellen Sie sicher, dass sich die neuen Werte einer migrierten Sequenz nicht mit den vorhandenen Werten überschneiden.

Um konvertierte Sequenzen mit dem letzten Wert aus der Quelldatenbank zu füllen

1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quelle.

2. Wählen Sie Einstellungen und dann Konvertierungseinstellungen.
3. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle nach PostgreSQL an.
4. Wählen Sie Konvertierte Sequenzen mit dem letzten auf der Quellseite generierten Wert auffüllen.
5. Wählen Sie „OK“, um die Einstellungen zu speichern und das Dialogfeld mit den Konvertierungseinstellungen zu schließen.

Konvertieren von Oracle ROWID

In einer Oracle-Datenbank enthält die ROWID-Pseudospalte die Adresse der Tabellenzeile. Die ROWID-Pseudospalte ist einzigartig für Oracle und AWS SCT konvertiert daher die ROWID-Pseudospalte in eine Datenspalte in PostgreSQL. Mit dieser Konvertierung können Sie die ROWID-Informationen beibehalten.

Bei der Konvertierung der ROWID-Pseudospalte AWS SCT kann eine Datenspalte mit dem Datentyp erstellt werden. `bigint` Wenn kein Primärschlüssel vorhanden ist, wird die ROWID-Spalte als Primärschlüssel AWS SCT festgelegt. Wenn ein Primärschlüssel vorhanden ist, wird für die ROWID-Spalte eine eindeutige Einschränkung AWS SCT festgelegt.

Wenn Ihr Quelldatenbankcode Operationen mit ROWID enthält, die Sie nicht mit einem numerischen Datentyp ausführen können, AWS SCT können Sie eine Datenspalte mit dem Datentyp erstellen. `character varying`

So erstellen Sie eine Datenspalte für Oracle ROWID für ein Projekt

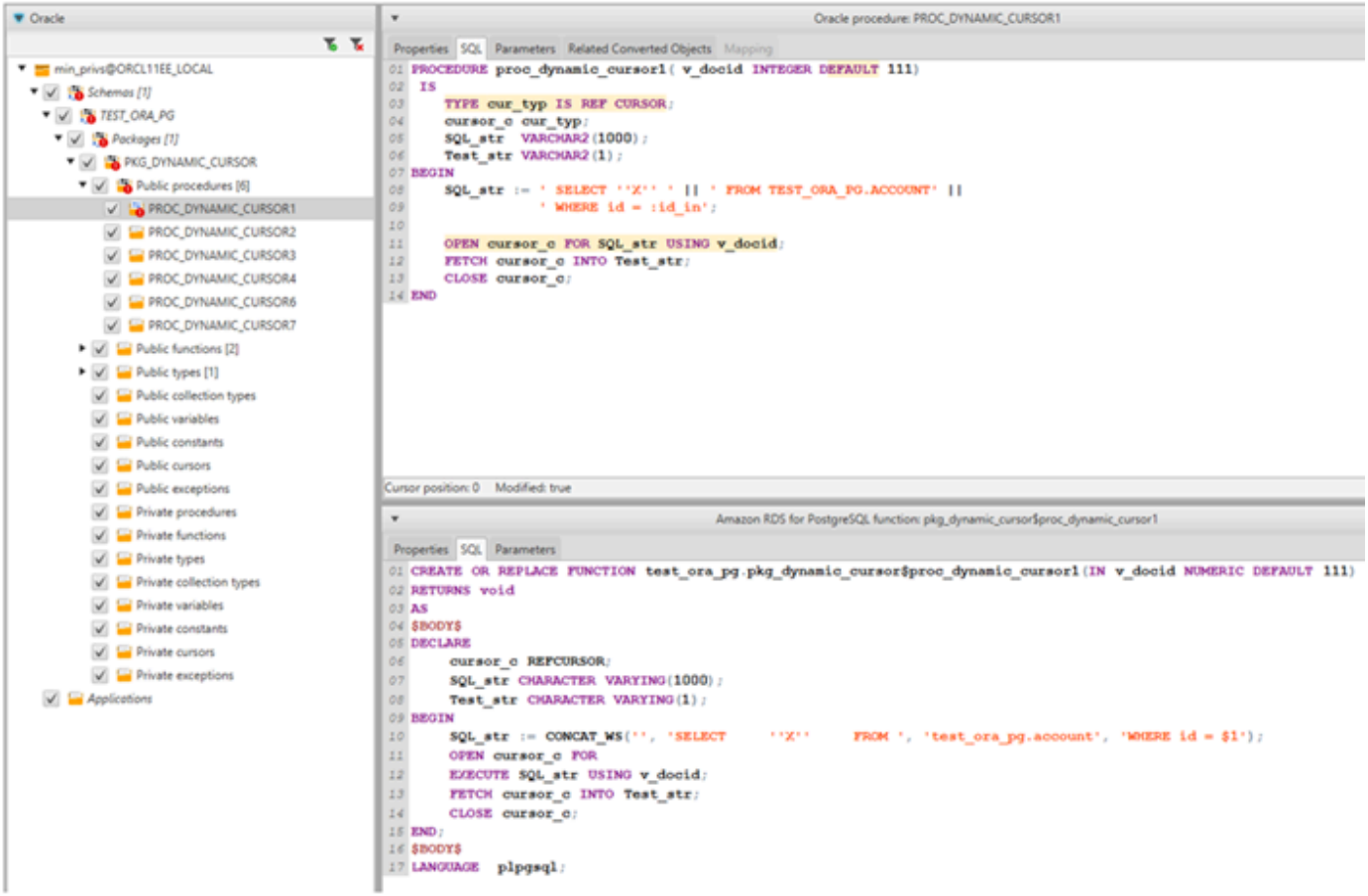
1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quelle.
2. Wählen Sie Einstellungen und dann Konvertierungseinstellungen.
3. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle nach PostgreSQL an.
4. Führen Sie für Generate row ID einen der folgenden Schritte aus:
   - Wählen Sie Als Identität generieren aus, um eine numerische Datenspalte zu erstellen.
   - Wählen Sie Als Zeichendomänentyp generieren, um eine Zeichendatenspalte zu erstellen.
5. Wählen Sie OK, um die Einstellungen zu speichern und das Dialogfeld mit den Konvertierungseinstellungen zu schließen.

Oracle Dynamic SQL wird konvertiert

Oracle bietet zwei Möglichkeiten, dynamisches SQL zu implementieren: die Verwendung einer EXECUTE IMMEDIATE-Anweisung oder das Aufrufen von Prozeduren im Paket DBMS_SQL. Wenn Ihre Oracle-Quelldatenbank Objekte mit dynamischem SQL enthält, verwenden Sie diese, AWS SCT um dynamische SQL-Anweisungen von Oracle nach PostgreSQL zu konvertieren.

Um Oracle Dynamic SQL nach PostgreSQL zu konvertieren

1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quelle.
2. Wählen Sie in der Oracle-Quellstrukturansicht ein Datenbankobjekt, das dynamisches SQL verwendet.
3. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt, wählen Sie Schema konvertieren und stimmen Sie zu, die Objekte zu ersetzen, falls sie existieren. Der folgende Screenshot zeigt die konvertierte Prozedur unterhalb der Oracle-Prozedur mit dynamischem SQL.



Oracle-Partitionen konvertieren

AWS SCT unterstützt derzeit die folgenden Partitionierungsmethoden:

- Bereich
- Auflisten
- Mehrspaltiger Bereich
- Hash
- Zusammengesetzt (Listenliste, Bereichsliste, Listenbereich, Listen-Hash, Bereichs-Hash, Hash-Hash)

Migration von Oracle zu Amazon RDS for MySQL oder Amazon Aurora MySQL mit dem AWS Schema Conversion Tool

Um Oracle-Datenbankfunktionen in Ihrem konvertierten MySQL-Code zu emulieren, verwenden Sie das Oracle to MySQL Extension Pack in AWS SCT. Weitere Informationen zu Erweiterungspaketen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Themen

- Rechte für MySQL als Zieldatenbank
- Einstellungen für die Konvertierung von Oracle in MySQL
- Überlegungen zur Migration
- Konvertierung der WITH-Anweisung in Oracle nach RDS für MySQL oder Amazon Aurora MySQL

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *

- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- TEMPORÄRE TABELLEN ERSTELLEN AUF * . *
- AWS_LAMBDA_ZUGANG
- EINFÜGEN, AUF AWS_ORACLE _EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_ORACLE _EXT_DATA. *

Wenn Sie eine MySQL-Datenbank der Version 5.7 oder niedriger als Ziel verwenden, gewähren Sie die INVOKE LAMBDA *.*-Berechtigung anstelle von _ACCESS. AWS_LAMBDA Erteilen Sie für MySQL-Datenbanken der Version 8.0 und höher die AWS_LAMBDA_ACCESS Erlaubnis.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Wenn Sie eine MySQL-Datenbank Version 5.7 oder niedriger als Ziel verwenden, verwenden Sie `GRANT INVOKE LAMBDA ON *.* TO 'user_name'` stattGRANT AWS_LAMBDA_ACCESS TO 'user_name'.

Setzen Sie den Parameter `lower_case_table_names` auf `1`, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf `1`. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Einstellungen für die Konvertierung von Oracle in MySQL

Um die Konvertierungseinstellungen von Oracle in MySQL zu bearbeiten AWS SCT, wählen Sie Einstellungen in und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — MySQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle nach MySQL an.

Die Konvertierungseinstellungen von Oracle nach MySQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um dem entgegenzuwirken, kann Ihre Oracle-Quelldatenbank die `ROWID` Pseudospalte verwenden, MySQL unterstützt jedoch keine ähnliche Funktionalität. AWS SCT kann die `ROWID`

Pseudospalte im konvertierten Code emulieren. Wählen Sie dazu unter Zeilen-ID generieren? die Option Als Identität generieren aus. .

Wenn Ihr Oracle-Quellcode die `ROWID` Pseudospalte nicht verwendet, wählen Sie Don't generate für Generate row ID? In diesem Fall arbeitet der konvertierte Code schneller.

- Um mit Ihrem Oracle-Quellcode zu arbeiten, wenn er die `TO_NUMBER` Funktionen `TO_CHAR``TO_DATE`, und mit Parametern enthält, die MySQL nicht unterstützt. AWS SCT Emuliert standardmäßig die Verwendung dieser Parameter im konvertierten Code.

  Wenn Ihr Oracle-Quellcode nur Parameter enthält, die PostgreSQL unterstützt, können Sie native MySQL- `TO_CHAR``TO_DATE`, und `TO_NUMBER` -Funktionen verwenden. In diesem Fall arbeitet der konvertierte Code schneller. Um nur diese Parameter einzubeziehen, wählen Sie die folgenden Werte aus:

  - Die Funktion TO_CHAR () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
  - Die Funktion TO_DATE () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
  - Die Funktion TO_NUMBER () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
- Um festzustellen, ob Ihre Datenbank und Anwendungen in unterschiedlichen Zeitzonen ausgeführt werden. AWS SCT Emuliert standardmäßig Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. In diesem Fall entspricht die Option Zeitzone auf der Clientseite der Zeitzone auf dem Server.

Überlegungen zur Migration

Wenn Sie Oracle in RDS for MySQL oder Aurora MySQL konvertieren, können Sie eine `GOTO` Anweisung und ein Label verwenden, um die Reihenfolge zu ändern, in der Anweisungen ausgeführt werden. Alle PL/SQL-Anweisungen, die auf eine `GOTO` Anweisung folgen, werden übersprungen, und die Verarbeitung wird an der Bezeichnung fortgesetzt. Sie können `GOTO` Anweisungen und Labels an einer beliebigen Stelle innerhalb einer Prozedur, eines Batches oder eines Anweisungsblocks verwenden. Sie können auch als Nächstes GOTO-Anweisungen verwenden.

MySQL verwendet keine `GOTO` Anweisungen. Wenn Code AWS SCT konvertiert wird, der eine `GOTO` Anweisung enthält, wird die Anweisung so konvertiert, dass sie eine `BEGIN…END LOOP…END LOOP` OR-Anweisung verwendet.

In der folgenden Tabelle finden Sie Beispiele dafür, wie `GOTO` Anweisungen AWS SCT konvertiert werden.

Oracle-Anweisung	MySQL-Anweisung

`BEGIN`
`  ....`
`  statement1;`
`  ....`
`  GOTO label1;`
`  statement2;`
`  ....`
`  label1:`
`  Statement3;`
`  ....`
`END`
	`BEGIN`
	` label1:`
	` BEGIN`
	`  ....`
	`  statement1;`
	`  ....`
	`  LEAVE label1;`
	`  statement2;`
	`  ....`
	` END;`
	`  Statement3;`
	`  ....`
	`END`

`BEGIN`
`  ....`
`  statement1;`
`  ....`
`  label1:`
`  statement2;`
`  ....`
`  GOTO label1;`
`  statement3;`
`  ....`
`  statement4;`
`  ....`
`END`
	`BEGIN`
	`  ....`
	`  statement1;`
	`  ....`
	`  label1:`
	`  LOOP`
	`  statement2;`
	`  ....`
	`  ITERATE label1;`
	`  LEAVE label1;`
	`  END LOOP;`
	`  statement3;`
	`  ....`
	`  statement4;`
	`  ....`
	`END`

`BEGIN`
`  ....`
`  statement1;`
`  ....`
`  label1:`
`  statement2;`
	`BEGIN`
	`  ....`
	`  statement1;`
	`  ....`
	`  label1:`
	`  BEGIN`

Oracle-Anweisung	MySQL-Anweisung
``` .... statement3; .... statement4; .... END ```	``` statement2; .... statement3; .... statement4; .... END; END ```

Konvertierung der WITH-Anweisung in Oracle nach RDS für MySQL oder Amazon Aurora MySQL

Sie verwenden die WITH-Klausel (subquery_factoring) in Oracle, um einem Unterabfragenblock einen Namen (query_name) zuzuweisen. Sie können dann an mehreren Stellen in der Abfrage auf den Unterabfragenblock Bezug nehmen. Wenn ein Unterabfrageblock keine Links oder Parameter (lokal, Prozedur, Funktion, Paket) enthält, wird die Klausel in eine Ansicht oder eine temporäre Tabelle AWS SCT konvertiert.

Der Vorteil der Umwandlung der Klausel in eine temporäre Tabelle besteht darin, dass wiederholte Verweise auf die Unterabfrage effizienter sein können. Die größere Effizienz liegt darin, dass die Daten leicht aus der temporären Tabelle abgerufen werden können, anstatt von jeder Referenz benötigt zu werden. Sie können dies emulieren, indem Sie zusätzliche Ansichten oder eine temporäre Tabelle verwenden. Für den Ansichtsnamen wird das Format `<procedure_name>$<subselect_alias>` verwendet.

Beispiele finden Sie in der folgenden Tabelle.

Oracle-Anweisung	MySQL-Anweisung

``
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
``
	``
	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);
	``

Oracle-Anweisung	MySQL-Anweisung

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

```
    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
        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;
```

Oracle-Anweisung	MySQL-Anweisung

```
CREATE PROCEDURE
 TEST_ORA_PG.P_WITH_SELECT_R
EGULAR_MULT_01
AS
BEGIN

 FOR cur IN  (
               WITH dept_empl 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_empl
.*,dept.*
                 FROM dept_empl, dept
                 WHERE dept_empl
.dept_id = dept.deptid
              ) LOOP
              NULL;
            END LOOP;
```

```
CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$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 = 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$deptname VARCHAR
 (200);
    DECLARE done INT DEFAULT FALSE;
    DECLARE cur CURSOR FOR SELECT
        dept_empl.*, dept.*
        FROM TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_empl
          ` AS dept_empl,
            TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
          ` AS dept
```

Oracle-Anweisung	MySQL-Anweisung
	WHERE dept_empl.DEPT_ID = dept.DEPTID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur; read_label: LOOP FETCH cur INTO varID, varNAME, 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_REGULAR_MULT_01()

Oracle-Anweisung	MySQL-Anweisung

```
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;
```

```
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:
```

Oracle-Anweisung	MySQL-Anweisung
	LOOP FETCH cur INTO var$ID; IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; END;

Migration von Oracle Database zu Amazon RDS for Oracle mit AWS Schema Conversion Tool

Bei der Migration von Oracle-Schema und -Code in Amazon RDS für Oracle sind einige Punkte zu beachten:

- AWS SCT kann Verzeichnisobjekte zum Objektbaum hinzufügen. Verzeichnisobjekte sind logische Strukturen, die jeweils ein physisches Verzeichnis im Dateisystem des Servers darstellen. Sie können Verzeichnisobjekte mit Paketen, wie z. B. DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, dem DATAPUMP Dienstprogramm usw. verwenden.
- AWS SCT unterstützt die Konvertierung von Oracle-Tablespaces in eine Amazon RDS for Oracle Oracle-DB-Instance. Oracle speichert Daten logisch in Tabellenräumen und physisch in Datendateien, die dem entsprechenden Tabellenraum zugeordnet sind. In Oracle können Sie einen Tabellenraum mit Datendateinamen erstellen. Amazon RDS unterstützt Oracle Managed Files (OMF) nur für Datendateien, Protokolldateien und Kontrolldateien. AWS SCT erstellt die benötigten Datendateien während der Konvertierung.
- AWS SCT kann Rollen und Rechte auf Serverebene konvertieren. Die Oracle-Datenbank-Engine verwendet rollenbasierte Sicherheit. Eine Rolle ist eine Sammlung von Sonderrechten, die Sie einem Benutzer gewähren oder entziehen können. Eine vordefinierte Rolle in Amazon RDS mit den Namen DBA lässt normalerweise alle administrativen Sonderrechte in einer Oracle-Datenbank-Engine zu. Die folgenden Sonderrechte sind nicht für die Feedback-Schleifen-Rolle in einer Amazon RDS-DB-Instance verfügbar, welche die Oracle-Engine verwendet:

- Ändern der Datenbank
- Ändern des Systems
- Erstellen eines Verzeichnisses
- Gewähren von Sonderrechten
- Gewähren einer Rolle
- Erstellen eines externen Auftrags

Sie können alle anderen Berechtigungen für eine Amazon RDS for Oracle-Benutzerrolle, u. a. erweiterte Filterungs- und Spaltenberechtigungen, erteilen.

- AWS SCT unterstützt die Konvertierung von Oracle-Jobs in Jobs, die auf Amazon RDS for Oracle ausgeführt werden können. Es gibt einige Einschränkungen für die Konvertierung, unter anderem:
  - Ausführbare Aufträge werden nicht unterstützt.
  - Die Einplanung von Aufträgen, die den ANYDATA-Datentyp als Argument verwenden, werden nicht unterstützt.
- Oracle Real Application Clusters (RAC) One Node ist eine Option der Oracle Database Enterprise Edition, die mit Oracle Database 11g Release 2 eingeführt wurde. Amazon RDS for Oracle unterstützt die RAC-Funktion nicht. Verwenden Sie Amazon RDS Multi-AZ für Hochverfügbarkeit.

  Bei einer Multi-AZ-Bereitstellung sorgt Amazon RDS für eine automatische Bereitstellung und Verwaltung eines synchronen Standby-Replikats in einer anderen Availability Zone. Die primäre DB-Instance wird über die Availability Zone synchron auf eine Standby-Replikation repliziert. Diese Funktionalität sorgt für Datenredundanz, vermeidet das Einfrieren von E/A-Vorgängen und minimiert Latenzspitzen während der Durchführung von Systemsicherungen.
- Oracle Spatial stellt ein SQL-Schema und Funktionen bereit, die Speicherung, Abruf, Update und Abfrage von Sammlungen großer Datensätze in einer Oracle-Datenbank ermöglichen. Oracle Locator stellt Funktionalitäten zur Verfügung, die üblicherweise für die Unterstützung von Anwendungen auf Internet- und Wireless-Servicebasis sowie von GIS-Lösungen auf Partnerbasis erforderlich sind. Oracle Locator ist ein begrenzter Teilbereich von Oracle Spatial.

  Um Oracle Spatial- und Oracle Locator-Funktionen verwenden zu können, fügen Sie die Option SPATIAL oder die Option LOCATOR (schließen sich gegenseitig aus) zur Optionsgruppe Ihrer DB-Instance hinzu.

  Es gibt einige Voraussetzungen für die Verwendung von Oracle Spatial und Oracle Locator auf einer Amazon RDS for Oracle-DB-Instance:

  - Die Instance sollte Oracle Enterprise Edition Version 12.1.0.2.v6 oder höher oder 11.2.0.4.v10 oder höher verwenden.
  - Die Instance sollte sich innerhalb einer Virtual Private Cloud (VPC) befinden.
  - Die Instance sollte die Version der DB-Instance-Klasse sein, die die Oracle-Funktion unterstützen kann. Beispiel: Oracle Spatial wird für die DB-Instance-Klassen db.m1.small, db.t1.micro, db.t2.micro und db.t2.small nicht unterstützt. Weitere Informationen finden Sie unter DB-Instance-Klassenunterstützung für Oracle.
  - Die Instance muss die Option „Auto Minor Version Upgrade“ (Upgrade einer Unterversion automatisch durchführen) aktiviert haben. Amazon RDS aktualisiert Ihre DB-Instance auf die neueste Oracle-PSU, falls Schwachstellen mit einem CVSS-Schweregrad von 9+ oder andere gemeldete Schwachstellen vorliegen. Weitere Informationen finden Sie unter

    Einstellungen für Oracle-DB-Instances.
  - Wenn Ihre DB-Instance Version 11.2.0.4.v10 oder höher ist, müssen Sie die XMLDB-Option installieren. Weitere Informationen finden Sie unter

    Oracle XML DB.
  - Sie sollten über eine Oracle Spatial-Lizenz von Oracle verfügen. Weitere Informationen finden Sie unter Oracle Spatial and Graph in der Oracle-Dokumentation.
- Data Guard ist in der Oracle Database Enterprise Edition enthalten. Verwenden Sie Amazon RDS Multi-AZ für Hochverfügbarkeit.

  Bei einer Multi-AZ-Bereitstellung sorgt Amazon RDS für eine automatische Bereitstellung und Verwaltung eines synchronen Standby-Replikats in einer anderen Availability Zone. Die primäre DB-Instance wird über die Availability Zone synchron auf eine Standby-Replikation repliziert. Diese Funktionalität sorgt für Datenredundanz, vermeidet das Einfrieren von E/A-Vorgängen und minimiert Latenzspitzen während der Durchführung von Systemsicherungen.
- AWS SCT unterstützt die Konvertierung von Oracle DBMS_SCHEDULER-Objekten bei der Migration zu Amazon RDS for Oracle. Der AWS SCT Bewertungsbericht gibt an, ob ein Zeitplanobjekt konvertiert werden kann. Weitere Informationen zur Verwendung von Zeitplanobjekten mit Amazon RDS finden Sie in der Amazon RDS-Dokumentation.
- Für Umwandlungen von Oracle zu Amazon RDS wird DB Links unterstützt. Ein Datenbanklink ist ein Schemaobjekt in einer Datenbank, mit dessen Hilfe Sie auf Objekte in einer anderen Datenbank zugreifen können. Die andere Datenbank muss keine Oracle-Datenbank sein. Für den Zugriff auf andere als Oracle-Datenbanken müssen Sie jedoch Oracle Heterogeneous Services verwenden.

Nachdem Sie einen Datenbanklink erstellt haben, können Sie den Link in SQL-Anweisungen verwenden, um auf Tabellen, Ansichten und PL/SQL-Objekte in der anderen Datenbank zu verweisen. Um einen Datenbanklink zu verwenden, fügen Sie dem Tabellen-, Ansicht- oder PL/SQL-Objektnamen `@dblink` hinzu. Sie können eine Tabelle oder Ansicht in der anderen Datenbank mit der SELECT-Anweisung abfragen. Weitere Informationen zur Verwendung von Oracle-Datenbanklinks finden Sie in der Oracle-Dokumentation.

Weitere Informationen zur Verwendung von Datenbanklinks mit Amazon RDS finden Sie in der Amazon RDS-Dokumentation.

- Der AWS SCT Bewertungsbericht enthält Servermetriken für die Konvertierung. Zu diesen Metriken über Ihre Oracle-Instance gehören:
  - Datenverarbeitungs- und Speicherkapazität der Ziel-DB-Instance.
  - Nicht unterstützte Oracle-Funktionen wie Real Application Clusters, die Amazon RDS nicht unterstützt.
  - Datenträger-Lese-/Schreib-Last
  - Durchschnittlicher Datenträger-Gesamtdurchsatz
  - Serverinformationen wie Servername, Betriebssystem, Hostname und Zeichensatz.

Rechte für RDS für Oracle als Ziel

Um zu Amazon RDS for Oracle zu migrieren, erstellen Sie einen privilegierten Datenbankbenutzer. Sie können das folgende Codebeispiel verwenden.

```
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;
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Einschränkungen bei der Konvertierung von Oracle zu Amazon RDS for Oracle

Bei der Migration von Oracle-Schema und -Code in Amazon RDS for Oracle sind einige Einschränkungen zu beachten:

- Eine vordefinierte Rolle in Amazon RDS mit den Namen DBA lässt normalerweise alle administrativen Sonderrechte in einer Oracle-Datenbank-Engine zu. Die folgenden Sonderrechte sind nicht für die Feedback-Schleifen-Rolle in einer Amazon RDS-DB-Instance verfügbar, welche die Oracle-Engine verwendet:
  - Ändern der Datenbank
  - Ändern des Systems

- Erstellen eines Verzeichnisses
- Gewähren von Sonderrechten
- Gewähren einer Rolle
- Erstellen eines externen Auftrags

Sie können einer Oracle RDS-Benutzerrolle alle anderen Berechtigungen erteilen.

- Amazon RDS for Oracle unterstützt traditionelle Prüfungen, detaillierte Prüfungen mit dem Paket DBMS_FGA und Oracle Unified Auditing.
- Amazon RDS for Oracle unterstützt Change Data Capture (CDC) nicht. Um CDC während und nach einer Datenbankmigration durchzuführen, verwenden Sie. AWS Database Migration Service

Herstellen einer Verbindung zu PostgreSQL-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode von PostgreSQL in die folgenden Ziele zu konvertieren:

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

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- Rechte für PostgreSQL als Quelldatenbank
- Verbindung zu PostgreSQL als Quelle herstellen
- Rechte für MySQL als Zieldatenbank

Rechte für PostgreSQL als Quelldatenbank

Die für PostgreSQL als Quelle erforderlichen Rechte lauten wie folgt:

- STELLEN SIE EINE VERBINDUNG ZUR DATENBANK HER *<database_name>*

- VERWENDUNG IM SCHEMA *<database_name>*
- WÄHLEN SIE FÜR ALLE TABELLEN IM SCHEMA *<database_name>*
- WÄHLEN SIE ALLE SEQUENZEN IM SCHEMA AUS *<database_name>*

Verbindung zu PostgreSQL als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer PostgreSQL-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer PostgreSQL-Quelldatenbank her

1. Wählen Sie im die Option AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie PostgreSQL und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen für die PostgreSQL-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein. Sie können mithilfe eines IPv6 Adressprotokolls eine Verbindung zu Ihrer PostgreSQL-Quelldatenbank herstelle

Parameter	Aktion
	n. Stellen Sie dazu sicher, dass Sie bei der Eingabe der IP-Adresse eckige Klammern verwenden, wie im folgenden Beispiel gezeigt. `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der PostgreSQL-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie nach dem Passwort gefragt, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustel len.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Vertrauensspeichers zu überprüfen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Speicherort im Bereich Allgemeine Einstellungen angezeigt wird, müssen Sie ihn hinzufügen.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank passwort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.
PostgreSQL-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Die Rechte, die für MySQL als Ziel erforderlich sind, wenn Sie von PostgreSQL migrieren, lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- EINFÜGEN, AUF AWS_POSTGRESQL _EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_POSTGRESQL _EXT_DATA. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_POSTGRESQL _EXT_DATA. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Setzen Sie den Parameter `lower_case_table_names` auf 1, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf 1. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Herstellen einer Verbindung zu SAP-Datenbanken mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode aus SAP (Sybase) Adaptive Server Enterprise (ASE) in die folgenden Ziele zu konvertieren:

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

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- Rechte für SAP ASE als Quelldatenbank
- Verbindung zu SAP ASE (Sybase) als Quelle herstellen

- Rechte für MySQL als Zieldatenbank
- Einstellungen für die Konvertierung von SAP ASE nach MySQL
- Rechte für PostgreSQL als Zieldatenbank
- Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL

Rechte für SAP ASE als Quelldatenbank

Um eine SAP ASE-Datenbank als Quelle zu verwenden, erstellen Sie einen Datenbankbenutzer und gewähren Berechtigungen. Gehen Sie dazu wie folgt vor.

Erstellen und konfigurieren Sie einen Datenbankbenutzer

1. Stellt eine Verbindung zur Quelldatenbank her.
2. Erstellen Sie einen Datenbankbenutzer mit den folgenden Befehlen. Geben Sie ein Passwort für den neuen Benutzer ein.

```
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. Erteilen Sie für jede Datenbank, die Sie migrieren möchten, die folgenden Rechte.

```
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
```

Verbindung zu SAP ASE (Sybase) als Quelle herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihrer SAP ASE-Quelldatenbank mit dem herzustellen AWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer SAP ASE-Quelldatenbank her

1. Wählen Sie in der die AWS Schema Conversion Tool Option Quelle hinzufügen aus.
2. Wählen Sie SAP ASE und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur SAP ASE-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der SAP ASE-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.

Parameter	Aktion
	Note AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgege ben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgefordert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustel len.
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Vertrauensspeichers zu überprüfen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank passwort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.

Parameter	Aktion
SAP ASE-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- EINFÜGEN, AUF AWS_SAPASE _EXT AKTUALISIEREN. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_SAPASE _EXT. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Setzen Sie den Parameter `lower_case_table_names` auf `1`, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf `1`. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Einstellungen für die Konvertierung von SAP ASE nach MySQL

Um die Einstellungen für die Konvertierung von SAP ASE in MySQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste SAP ASE und dann SAP ASE — MySQL oder SAP ASE — Amazon Aurora (MySQL-kompatibel) aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL an.

Die Einstellungen für die Konvertierung von SAP ASE in MySQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Um die exakten Namen der Quelldatenbankobjekte im konvertierten Code zu verwenden.

  AWS SCT Konvertiert standardmäßig die Namen von Datenbankobjekten, Variablen und Parametern in Kleinbuchstaben. Um die ursprüngliche Groß- und Kleinschreibung für diese Namen beizubehalten, wählen Sie Objektnamen der Quelldatenbank unter Berücksichtigung von Groß- und Kleinschreibung behandeln aus. Wählen Sie diese Option, wenn Sie in Ihrem SAP ASE-Quelldatenbankserver Objektnamen verwenden, bei denen Groß- und Kleinschreibung beachtet wird.

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *`user_name`* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *`db_name`* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *`your_password`* durch ein sicheres Passwort.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL

Um die Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste SAP ASE und dann SAP ASE — PostgreSQL oder SAP ASE — Amazon Aurora (PostgreSQL-kompatibel) aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL an.

Die Einstellungen für die Konvertierung von SAP ASE nach PostgreSQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die Vorlage zu definieren, die für die Schemanamen im konvertierten Code verwendet werden soll. Wählen Sie für die Vorlage zur Generierung von Schemanamen eine der folgenden Optionen aus:
  - <source_db>— Verwendet den SAP ASE-Datenbanknamen als Schemanamen in PostgreSQL.
  - <source_schema>— Verwendet den SAP ASE-Schemanamen als Schemanamen in PostgreSQL.
  - _ <source_db><schema>— Verwendet eine Kombination aus der SAP ASE-Datenbank und Schemanamen als Schemanamen in PostgreSQL.

- Um die exakten Namen der Quelldatenbankobjekte im konvertierten Code zu verwenden.

  AWS SCT Konvertiert standardmäßig die Namen von Datenbankobjekten, Variablen und Parametern in Kleinbuchstaben. Um die ursprüngliche Groß- und Kleinschreibung für diese Namen beizubehalten, wählen Sie Objektnamen der Quelldatenbank unter Berücksichtigung von Groß- und Kleinschreibung behandeln aus. Wählen Sie diese Option, wenn Sie in Ihrem SAP ASE-Quelldatenbankserver Objektnamen verwenden, bei denen Groß- und Kleinschreibung beachtet wird.

  Bei Operationen, bei denen die Groß- und Kleinschreibung beachtet wird, AWS SCT kann die Konvertierung von Datenbankobjektnamen in Kleinbuchstaben vermieden werden. Wählen Sie dazu bei Operationen, bei denen die Groß- und Kleinschreibung beachtet wird, die Umwandlung in Kleinbuchstaben vermeiden aus.
- Um die Verwendung von Indizes mit demselben Namen in verschiedenen Tabellen in SAP ASE zu ermöglichen.

  In PostgreSQL müssen alle Indexnamen, die Sie im Schema verwenden, eindeutig sein. Um sicherzustellen, dass dadurch eindeutige Namen für alle Ihre Indizes AWS SCT generiert werden, wählen Sie Eindeutige Namen für Indizes generieren aus.

Connect Microsoft SQL Server mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Datenbankcode-Objekte und Anwendungscode aus SQL Server in die folgenden Ziele zu konvertieren:

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

> **Note**
>
> AWS SCT unterstützt nicht die Verwendung von Amazon RDS for SQL Server als Quelle.

Sie können ihn verwenden AWS SCT , um einen Bewertungsbericht für die Migration von Schemas, Datenbankcodeobjekten und Anwendungscode von SQL Server zu Babelfish for Aurora PostgreSQL zu erstellen, wie im Folgenden beschrieben.

Themen

- Berechtigungen für Microsoft SQL Server als Quelle
- Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle
- Verbindung zu SQL Server als Quelle herstellen
- Umwandeln von SQL Server in MySQL
- Migration von SQL Server zu PostgreSQL mit AWS Schema Conversion Tool
- Migration von SQL Server zu Amazon RDS for SQL Server mit AWS Schema Conversion Tool

Berechtigungen für Microsoft SQL Server als Quelle

Die für Microsoft SQL Server als Quelle erforderlichen Rechte lauten wie folgt:

- VIEW DEFINITION
- VIEW DATABASE STATE

Mit dieser `VIEW DEFINITION` Berechtigung können Benutzer mit öffentlichem Zugriff Objektdefinitionen einsehen. AWS SCT verwendet das `VIEW DATABASE STATE` Recht, um die Funktionen der SQL Server Enterprise Edition zu überprüfen.

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Gewähren Sie außerdem die folgenden Berechtigungen für die Datenbank `master`:

- VIEW SERVER STATE
- VIEW ANY DEFINITION

AWS SCT verwendet das `VIEW SERVER STATE` Recht, Servereinstellungen und -konfigurationen zu sammeln. Stellen Sie sicher, dass Sie die `VIEW ANY DEFINITION` Berechtigung zum Anzeigen von Endpunkten gewähren.

Führen Sie den folgenden Befehl in der Datenbank `master` aus, um Informationen über Microsoft Analysis Services zu lesen.

```
EXEC master..sp_addsrvrolemember @loginame = N'<user_name>', @rolename = N'sysadmin'
```

Ersetzen Sie im vorherigen Beispiel den *<user_name>* Platzhalter durch den Namen des Benutzers, dem Sie zuvor die Berechtigungen erteilt haben.

Um Informationen über den SQL Server-Agent zu lesen, fügen Sie Ihren Benutzer zur `SQLAgentUser` Rolle hinzu. Führen Sie den folgenden Befehl für die Datenbank `msdb` aus.

```
EXEC sp_addrolemember <SQLAgentRole>, <user_name>;
```

Ersetzen Sie im vorherigen Beispiel den Platzhalter *<SQLAgentRole>* durch den Namen der SQL-Server-Agent-Rolle. Ersetzen Sie dann den *<user_name>* Platzhalter durch den Namen des Benutzers, dem Sie zuvor die Berechtigungen erteilt haben. Weitere Informationen finden Sie unter Hinzufügen eines Benutzers zur SQLAgent Benutzerrolle im Amazon RDS-Benutzerhandbuch.

Gewähren Sie die Berechtigung `SELECT on dbo.log_shipping_primary_databases` für die Datenbank `msdb`, um den Versand von Protokollen zu erkennen.

Um den Benachrichtigungsansatz der DDL-Replikation zu verwenden, gewähren Sie die entsprechenden `RECEIVE ON` *<schema_name>*.*<queue_name>* Rechte für Ihre Quelldatenbanken. Ersetzen Sie in diesem Beispiel den Platzhalter *<schema_name>* durch den Schemanamen Ihrer Datenbank. Ersetzen Sie dann den Platzhalter *<queue_name>* durch den Namen einer Warteschlangen-Tabelle.

Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle

Wenn Ihre Anwendung auf einem Windows-basierten Intranet ausgeführt wird, können Sie möglicherweise die Windows-Authentifizierung für den Zugriff auf die Datenbank verwenden. Die Windows-Authentifizierung verwendet die aktuelle Windows-Identität auf der Betriebssystem-Thread für den Zugriff auf die SQL Server-Datenbank. Sie können dann die Windows-Identität auf eine SQL Server-Datenbank und Berechtigungen abbilden. Zum Herstellen einer Verbindung mit SQL Server unter Verwendung der Windows-Authentifizierung müssen Sie die Windows-Identität angeben, die Ihre Anwendung verwendet. Außerdem müssen Sie der Windows-Identität Zugriff auf die SQL Server-Datenbank erteilen.

SQL Server verfügt über zwei Zugriffsmodi: Windows-Authentifizierung-Modus und gemischter Modus. Der Windows-Authentifizierung-Modus aktiviert die Windows-Authentifizierung und deaktiviert die SQL Server-Authentifizierung. Der gemischte Modus aktiviert die Windows-Authentifizierung und die SQL Server-Authentifizierung. Die Windows-Authentifizierung ist immer verfügbar und kann nicht deaktiviert werden. Weitere Informationen über die Windows-Authentifizierung finden Sie in der Microsoft Windows-Dokumentation.

Nachfolgend finden Sie ein Beispiel für das Erstellen eines Benutzers in TEST_DB.

```
USE [TEST_DB]
CREATE USER [TestUser] FOR LOGIN [TestDomain\TestUser]
GRANT VIEW DEFINITION TO [TestUser]
GRANT VIEW DATABASE STATE TO [TestUser]
```

Verwenden der Windows-Authentifizierung mit einer JDBC-Verbindung

Der JDBC-Treiber unterstützt keine Windows-Authentifizierung, wenn der Treiber für Nicht-Windows-Betriebssysteme verwendet wird. Anmeldeinformationen für die Windows-Authentifizierung, wie z. B. Benutzername und Kennwort, werden nicht automatisch angegeben, wenn von einem anderen Betriebssystem als Windows aus eine Verbindung zu SQL Server hergestellt wird. In solchen Fällen müssen die Anwendungen stattdessen die SQL Server-Authentifizierung verwenden.

In der JDBC-Verbindungszeichenfolge, muss der Parameter `integratedSecurity` angegeben werden, um eine Verbindung unter Verwendung der Windows-Authentifizierung herzustellen. Der JDBC-Treiber unterstützt die integrierte Windows-Authentifizierung auf Windows-Betriebssysteme über den `integratedSecurity`-Verbindungszeichenfolgenparameter.

Verwendung der integrierten Authentifizierung

1. Installieren Sie den JDBC-Treiber.
2. Kopieren Sie die `sqljdbc_auth.dll`-Datei in ein Verzeichnis im Windows-Systempfad auf dem Computer, auf dem der JDBC-Treiber installiert ist.

   Die `sqljdbc_auth.dll` Dateien werden am folgenden Speicherort installiert:

   <Installationsverzeichnis>\sqljdbc_<version>\<language>\auth\

Wenn Sie versuchen, eine Verbindung zu SQL Server-Datenbank unter Verwendung der Windows-Authentifizierung herzustellen, ist es möglich, dass der folgende Fehler auftritt: Dieser Treiber ist nicht

für die integrierte Authentifizierung konfiguriert. Dieses Problem kann gelöst werden, indem Sie die folgenden Aktionen ausführen:

- Deklarieren Sie zwei Variablen, die auf den installierten Pfad Ihres JDBC verweisen:

  `variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu`(wo Ihre sqljdbc4.jar existiert);

  `variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu\auth\x86`(wenn Sie ein 32-Bit-Betriebssystem verwenden) oder `D\lib\JDBC4.1\enu\auth\x64` (wenn Sie ein 64-Bit-Betriebssystem verwenden). Hier befindet sich Ihr`sqljdbc_auth.dll`.

- Kopieren Sie `sqljdbc_auth.dll` in den Ordner, in dem Ihr JDK/JRE läuft. Sie können in den lib-Ordner, in den bin-Ordner usw. kopieren. Beispielsweise können Sie in den folgenden Ordner kopieren.

  ```
  [JDK_INSTALLED_PATH]\bin;
  [JDK_INSTALLED_PATH]\jre\bin;
  [JDK_INSTALLED_PATH]\jre\lib;
  [JDK_INSTALLED_PATH]\lib;
  ```

- Stellen Sie sicher, dass Sie in Ihrem JDBC-Bibliotheksordner nur die JAR-Datei haben. SQLJDBC4 Entfernen Sie alle anderen sqljdbc*.jar-Dateien aus diesem Ordner (oder kopieren Sie sie in einen anderen Ordner). Wenn Sie den Treiber als Teil Ihres Programms hinzufügen, stellen Sie sicher, dass Sie nur SQLJDBC4 .jar als zu verwendenden Treiber hinzufügen.
- Kopieren Sie die sqljdbc_auth.dll-Datei in den Ordner mit Ihrer Anwendung.

> **Note**
>
> Wenn Sie eine 32-Bit-JVM (Java Virtual Machine) ausführen, verwenden Sie die sqljdbc_auth.dll-Datei im Ordner x86, auch wenn das Betriebssystem die x64-Version ist. Wenn Sie eine 64-Bit-JVM auf einem x64-Prozessor ausführen, verwenden Sie die sqljdbc_auth.dll-Datei im Ordner x64.

Wenn Sie eine Verbindung zu einer SQL Server-Datenbank herstellen, können Sie entweder Windows Authentication (Windows-Authentifizierung) oder SQL Server Authentication (SQL Server-Authentifizierung) für die Option Authentication (Authentifizierung) verwenden.

Verbindung zu SQL Server als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Microsoft SQL Server-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Microsoft SQL Server-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen aus.
2. Wählen Sie Microsoft SQL Server und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unter Konfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Microsoft SQL Server-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein. Sie können mithilfe eines IPv6 Adressprotokolls eine Verbindung zu Ihrer SQL Server-Quelldatenbank herstelle n. Stellen Sie dazu sicher, dass Sie bei der Eingabe der IP-Adresse eckige Klammern verwenden, wie im folgenden Beispiel gezeigt.

Parameter	Aktion
	`[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Instance name	Geben Sie den Instance-Namen für die SQL-Server datenbank ein. Um den Instance-Namen abzurufen, senden Sie die Abfrage `SELECT @@servername;` an Ihre SQL-Serverdatenbank.
Authentifizierung	Wählen Sie den Authentifizierungstyp aus Windows-Authentifizierung und SQL Server-Authentifizierung aus.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT Projekt schließen und erneut öffnen, werden Sie zur Eingabe des Kennworts aufgefordert, um bei Bedarf eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat vertrauen: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Speicherort im Bereich Allgemeine Einstellungen angezeigt wird, stellen Sie sicher, dass Sie ihn hinzufügen.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank passwort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.
Sql Server Driver Path	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.
Bibliothek für die Windows-Authentifizierung	Geben Sie den Pfad zur `sqljdbc_auth.dll` Datei ein. Standardmäßig ist diese Datei am folgenden Speicherort installiert: *`<installation directory of the JDBC driver>`*`sqljdbc_`*`<version>`* `\`*`<language>`* `\auth\`

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.

6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Umwandeln von SQL Server in MySQL

Verwenden Sie das Erweiterungspaket SQL Server to MySQL, um Microsoft SQL Server-Datenbankfunktionen in Ihrem konvertierten MySQL-Code zu emulieren. AWS SCT Weitere Informationen zu Erweiterungspaketen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Themen

- Rechte für MySQL als Zieldatenbank
- Einstellungen für die Konvertierung von SQL Server in MySQL
- Überlegungen zur Migration

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- REFERENZEN AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- AUSLÖSEN AM * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- EINFÜGEN, AUF AWS_SQLSERVER _EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_SQLSERVER _EXT_DATA. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_SQLSERVER _EXT_DATA. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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';
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

Wenn Sie eine MySQL-Datenbank Version 5.7 oder niedriger als Ziel verwenden, führen Sie den folgenden Befehl aus. Für MySQL-Datenbanken der Version 8.0 und höher ist dieser Befehl veraltet.

```
GRANT SELECT ON mysql.proc TO 'user_name';
```

Setzen Sie den Parameter `lower_case_table_names` auf 1, um Amazon RDS für MySQL oder Aurora MySQL als Ziel zu verwenden. Dieser Wert bedeutet, dass der MySQL-Server Kennungen von Objektnamen wie Tabellen, Indizes, Auslösern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Ziel-Instance aktiviert haben, setzen Sie den Parameter `log_bin_trust_function_creators` auf 1. In diesem Fall müssen Sie die Eigenschaften `DETERMINISTIC`, `READS SQL DATA` oder `NO SQL` nicht verwenden, um gespeicherte Funktionen zu erstellen. Erstellen Sie zum Konfigurieren dieser Parameter eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Einstellungen für die Konvertierung von SQL Server in MySQL

Um die Einstellungen für die Konvertierung von SQL Server in MySQL zu bearbeiten, AWS SCT wählen Sie Einstellungen und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste

SQL Server und dann SQL Server — MySQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server zu MySQL an.

Die Einstellungen für die Konvertierung von SQL Server in MySQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Damit Ihre SQL Server-Quelldatenbank die Ausgabe von `EXEC` in einer Tabelle speichern kann. AWS SCT erstellt temporäre Tabellen und ein zusätzliches Verfahren, um diese Funktion zu emulieren. Um diese Emulation zu verwenden, wählen Sie Zusätzliche Routinen für den Umgang mit offenen Datensätzen erstellen aus.

Überlegungen zur Migration

Beachten Sie bei der Migration eines SQL Server-Schemas zu MySQL die folgenden Punkte:

- MySQL unterstützt die `MERGE` Anweisung nicht. AWS SCT Kann die `MERGE` Anweisung jedoch während der Konvertierung emulieren, indem es die `INSERT ON DUPLICATE KEY` Klausel und die `UPDATE FROM and DELETE FROM` Anweisungen verwendet.

  Stellen Sie für eine korrekte Verwendung der Emulation sicher`INSERT ON DUPLICATE KEY`, dass in der MySQL-Zieldatenbank eine eindeutige Einschränkung oder ein eindeutiger Primärschlüssel vorhanden ist.
- Sie können eine `GOTO` Anweisung und ein Label verwenden, um die Reihenfolge zu ändern, in der Anweisungen ausgeführt werden. Alle Transact-SQL-Anweisungen, die auf eine `GOTO` Anweisung folgen, werden übersprungen, und die Verarbeitung wird an der Bezeichnung fortgesetzt. Sie können `GOTO` Anweisungen und Labels an einer beliebigen Stelle innerhalb einer Prozedur, eines Batches oder eines Anweisungsblocks verwenden. Sie können `GOTO` Anweisungen auch verschachteln.

MySQL verwendet keine GOTO Anweisungen. Wenn Code AWS SCT konvertiert wird, der eine GOTO Anweisung enthält, wird die Anweisung so konvertiert, dass sie eine BEGIN…END LOOP…END LOOP OR-Anweisung verwendet. In der folgenden Tabelle finden Sie Beispiele dafür, wie GOTO Anweisungen AWS SCT konvertiert werden.

SQL Server-Anweisung	MySQL-Anweisung
`BEGIN .... statement1; .... GOTO label1; statement2; .... label1: Statement3; .... END`	`BEGIN label1: BEGIN .... statement1; .... LEAVE label1; statement2; .... END; Statement3; .... END`
`BEGIN .... statement1; .... label1: statement2; .... GOTO label1; statement3; .... statement4; .... END`	`BEGIN .... statement1; .... label1: LOOP statement2; .... ITERATE label1; LEAVE label1; END LOOP; statement3; .... statement4; .... END`

SQL Server-Anweisung	MySQL-Anweisung

`BEGIN`
`  ....`
`  statement1;`
`  ....`
`  label1:`
`  statement2;`
`  ....`
`  statement3;`
`  ....`
`  statement4;`
`  ....`
`END`
	`BEGIN`
	`  ....`
	`  statement1;`
	`  ....`
	`  label1:`
	`  BEGIN`
	`  statement2;`
	`  ....`
	`  statement3;`
	`  ....`
	`  statement4;`
	`  ....`
	`  END;`
	`END`

- MySQL unterstützt keine Tabellenwertfunktionen mit mehreren Anweisungen. AWS SCT simuliert Tabellenwertfunktionen während einer Konvertierung, indem temporäre Tabellen erstellt und Anweisungen so umgeschrieben werden, dass sie diese temporären Tabellen verwenden.

Migration von SQL Server zu PostgreSQL mit AWS Schema Conversion Tool

Sie können das Erweiterungspaket SQL Server to PostgreSQL in verwenden. AWS SCT Dieses Erweiterungspaket emuliert SQL Server-Datenbankfunktionen im konvertierten PostgreSQL-Code. Verwenden Sie das Erweiterungspaket SQL Server to PostgreSQL, um SQL Server Agent und SQL Server Database Mail zu emulieren. Weitere Informationen zu Erweiterungspaketen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Themen

- Rechte für PostgreSQL als Zieldatenbank
- Einstellungen für die Konvertierung von SQL Server in PostgreSQL
- Konvertieren von SQL Server-Partitionen in PostgreSQL Version 10-Partitionen
- Überlegungen zur Migration
- Verwenden eines AWS SCT Erweiterungspakets zum Emulieren des SQL Server-Agents in PostgreSQL

- Verwenden eines AWS SCT Erweiterungspakets zum Emulieren von SQL Server-Datenbank-Mail in PostgreSQL

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in. `"$user", public_synonyms, public`

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
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;
```

Ersetzen Sie ihn im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *db_name* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *your_password* durch ein sicheres Passwort.

In PostgreSQL kann nur der Schemaeigentümer oder ein `superuser` ein Schema entfernen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die Rolle `superuser`, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von SQL Server in PostgreSQL

Um die Einstellungen für die Konvertierung von SQL Server in PostgreSQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste SQL Server und dann SQL Server — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server zu PostgreSQL an.

Die Einstellungen für die Konvertierung von SQL Server in PostgreSQL AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die Verwendung von Indizes mit demselben Namen in verschiedenen Tabellen in SQL Server zu ermöglichen.

  In PostgreSQL müssen alle Indexnamen, die Sie im Schema verwenden, eindeutig sein. Um sicherzustellen, dass dadurch eindeutige Namen für alle Ihre Indizes AWS SCT generiert werden, wählen Sie Eindeutige Namen für Indizes generieren aus.
- Um SQL Server-Prozeduren in PostgreSQL-Funktionen zu konvertieren.

  PostgreSQL Version 10 und früher unterstützt keine Prozeduren. Kunden, die mit der Verwendung von Prozeduren in PostgreSQL nicht vertraut sind, AWS SCT können Prozeduren in Funktionen konvertieren. Wählen Sie dazu Verfahren in Funktionen konvertieren aus.
- Um die Ausgabe von `EXEC` in einer Tabelle zu emulieren.

  Ihre SQL Server-Quelldatenbank kann die Ausgabe von `EXEC` in einer Tabelle speichern. AWS SCT erstellt temporäre Tabellen und eine zusätzliche Prozedur, um diese Funktion zu emulieren. Um diese Emulation zu verwenden, wählen Sie Zusätzliche Routinen für den Umgang mit offenen Datensätzen erstellen aus.
- Um die Vorlage zu definieren, die für die Schemanamen im konvertierten Code verwendet werden soll. Wählen Sie für die Vorlage zur Generierung von Schemanamen eine der folgenden Optionen aus:
  - **<**source_db**>**— Verwendet den SQL Server-Datenbanknamen als Schemanamen in PostgreSQL.
  - **<**source_schema**>**— Verwendet den SQL Server-Schemanamen als Schemanamen in PostgreSQL.
  - _ <source_db><schema>— Verwendet eine Kombination aus der SQL Server-Datenbank und Schemanamen als Schemanamen in PostgreSQL.
- Um die Groß- und Kleinschreibung Ihrer Quellobjektnamen beizubehalten.

Um zu verhindern, dass Objektnamen in Kleinbuchstaben umgewandelt werden, wählen Sie bei Operationen, bei denen Groß- und Kleinschreibung beachtet wird, die Option Kleinschreibung vermeiden aus. Diese Option gilt nur, wenn Sie die Option zur Berücksichtigung von Groß- und Kleinschreibung in Ihrer Zieldatenbank aktivieren.

- Um die Parameternamen aus Ihrer Quelldatenbank beizubehalten.

  Um den Namen der Parameter im konvertierten Code doppelte Anführungszeichen hinzuzufügen, wählen Sie Ursprüngliche Parameternamen beibehalten.

Konvertieren von SQL Server-Partitionen in PostgreSQL Version 10-Partitionen

Beachten Sie Folgendes, wenn Sie eine Microsoft SQL Server-Datenbank in Amazon Aurora PostgreSQL-Compatible Edition (Aurora PostgreSQL) oder Amazon Relational Database Service for PostgreSQL (Amazon RDS for PostgreSQL) konvertieren.

In SQL Server können Sie Partitionen mit Partitionsfunktionen erstellen. Bei der Umwandlung einer partitionierten SQL Server-Tabelle in eine partitionierte PostgreSQL Version 10-Tabelle sind die folgenden potenziellen Probleme zu beachten:

- SQL Server ermöglicht es Ihnen, eine Tabelle mit einer Spalte ohne NOT NULL-Einschränkung zu partitionieren. In diesem Fall werden alle NULL-Werte in die ganz links befindliche Partition verschoben. PostgreSQL unterstützt keine NULL-Werte für RANGE-Partitionierungen.
- SQL Server ermöglicht Ihnen die Erstellung primärer und eindeutiger Schlüssel für partitionierte Tabellen. Sie erstellen für PostgreSQL primäre oder eindeutige Schlüssel direkt für jede Partition. Daher muss bei einer Migration in PostgreSQL die PRIMARY KEY- oder UNIQUE KEY-Einschränkung von ihrer übergeordneten Tabelle entfernt werden. Die `<original_key_name>_<partition_number>` resultierenden Schlüsselnamen haben das folgende Format.
- SQL Server ermöglicht Ihnen die Erstellung von Einschränkungen für Fremdschlüsselverweise aus und in partitionierte Tabellen. PostgreSQL unterstützt keine Fremdschlüssel, die auf partitionierte Tabellen verweisen. PostgreSQL unterstützt auch keine Fremdschlüsselverweise aus einer partitionierten Tabelle in eine andere Tabelle.
- SQL Server ermöglicht Ihnen die Erstellung von Indizes für partitionierte Tabellen. Für PostgreSQL sollte für jede Partition direkt ein Index erstellt werden. Folglich müssen Indizes bei der Migration in PostgreSQL aus den ihnen übergeordneten Tabellen entfernt werden. Die sich ergebenden Indexnamen weisen das Format `<original_index_name>_<partition_number>` auf.

- PostgreSQL unterstützt keine partitionierten Indizes.

Überlegungen zur Migration

Einige Dinge, die Sie bei der Migration eines SQL Server-Schemas nach PostgreSQL beachten sollten:

- In PostgreSQL müssen alle Objektnamen in einem Schema eindeutig sein. Dies gilt auch für Indizes. Indexnamen müssen im Schema der Basistabelle eindeutig sein. In SQL Server kann ein Indexname für verschiedene Tabellen gleich sein.

  Um die Eindeutigkeit von Indexnamen zu gewährleisten, AWS SCT haben Sie die Möglichkeit, eindeutige Indexnamen zu generieren, falls Ihre Indexnamen nicht eindeutig sind. Wählen Sie hierzu die Option Generate unique index names (Eindeutige Indexnamen generieren) in den Projekteigenschaften. Diese Funktion ist standardmäßig aktiviert. Wenn diese Option aktiviert ist, werden eindeutige Indexnamen im Format IX_table_name_index_name erstellt. Wenn diese Option deaktiviert ist, werden Indexnamen nicht geändert.
- Mithilfe einer GOTO-Anweisung und einer Bezeichnung lässt sich die Reihenfolge ändern, in der Anweisungen ausgeführt werden. Alle Transact-SQL-Anweisungen, die auf eine GOTO-Anweisung folgen, werden übersprungen, und die Verarbeitung wird ab der Bezeichnung fortgesetzt. GOTO-Anweisungen und Bezeichnungen können an beliebiger Stelle innerhalb eines Verfahrens, Stapels oder Anweisungsblocks verwendet werden. GOTO-Anweisungen können auch verschachtelt werden.

  PostgreSQL verwendet keine GOTO-Anweisungen. Wenn Code AWS SCT konvertiert wird, der eine GOTO-Anweisung enthält, wird die Anweisung so konvertiert, dass sie eine BEGIN... END- oder LOOP... END LOOP-Anweisung verwendet. In der folgenden Tabelle finden Sie Beispiele dafür, wie GOTO-Anweisungen AWS SCT konvertiert werden.

  SQL Server GOTO-Anweisungen und die konvertierten PostgreSQL-Anweisungen

SQL Server-Anweisung	PostgreSQL-Anweisung
`BEGIN .... statement1; .... GOTO label1; statement2;`	`BEGIN label1: BEGIN .... statement1; ....`

SQL Server-Anweisung	PostgreSQL-Anweisung
.... label1: Statement3; .... END	EXIT label1; statement2; .... END; Statement3; .... END
BEGIN .... statement1; .... label1: statement2; .... GOTO label1; statement3; .... statement4; .... END	BEGIN .... statement1; .... label1: LOOP statement2; .... CONTINUE label1; EXIT label1; END LOOP; statement3; .... statement4; .... END
BEGIN .... statement1; .... label1: statement2; .... statement3; .... statement4; .... END	BEGIN .... statement1; .... label1: BEGIN statement2; .... statement3; .... statement4; .... END; END

- PostgreSQL unterstützt keine MERGE-Anweisung. AWS SCT emuliert das Verhalten einer MERGE-Anweisung auf folgende Weise:
  - Durch INSERT ON CONFLICT-Konstruktion.
  - Durch die Verwendung der UPDATE FROM DML-Anweisung, wie z. B. MERGE ohne WHEN NOT MATCHED-Klausel.
  - Durch die Verwendung von CURSOR, z. B. mit einer MERGE-Anweisung mit DELETE-Klausel, oder durch die Verwendung einer komplexen MERGE ON-Bedingungsanweisung.
- AWS SCT kann Datenbank-Trigger zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- AWS SCT kann Trigger auf Serverebene zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- SQL Server erstellt `deleted` und verwaltet automatisch `inserted` Tabellen. Sie können diese temporären, speicherresidenten Tabellen verwenden, um die Auswirkungen bestimmter Datenänderungen zu testen und Bedingungen für DML-Triggeraktionen festzulegen. AWS SCT kann die Verwendung dieser Tabellen innerhalb von DML-Triggeranweisungen konvertieren.
- AWS SCT kann verknüpfte Server zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- Bei der Migration von Microsoft SQL Server zu PostgreSQL wird die integrierte SUSER_SNAME-Funktion wie folgt umgewandelt:
  - SUSER_SNAME – Gibt den Anmeldenamen zurück, der einer Sicherheits-ID (SID) zugeordnet ist.
  - SUSER_SNAME(<server_user_sid>) – Nicht unterstützt.
  - SUSER_SNAME() CURRENT_USER – Gibt den Benutzernamen des aktuellen Ausführungskontexts zurück.
  - SUSER_SNAME(NULL) – Gibt NULL zurück.
- Die Umwandlung von Tabellenwertfunktionen wird unterstützt. Tabellenwertfunktionen geben eine Tabelle zurück und können in einer Abfrage die Stelle einer Tabelle einnehmen.
- PATINDEX gibt die Startposition des ersten Auftretens eines Musters in einem bestimmten Ausdruck zurück. Dabei werden alle gültigen Text und Zeichendatentypen unterstützt. Nullen werden zurückgegeben, wenn das Muster nicht gefunden wird. AWS SCT <pattern character><expression character varying>Ersetzt bei der Konvertierung von SQL Server zu Amazon RDS for PostgreSQL den Anwendungscode, der PATINDEX verwendet, durch aws_sqlserver_ext.patindex (,).
- In SQL Server ist ein benutzerdefinierter Tabellentyp ein Typ, der die Definition einer Tabellenstruktur darstellt. Sie verwenden einen benutzerdefinierten Tabellentyp, um table-value-

Parameter für gespeicherte Prozeduren oder Funktionen zu deklarieren. Sie können auch einen benutzerdefinierten Tabellentyp verwenden, um Tabellenvariablen zu deklarieren, die Sie in einem Batch oder im Hauptteil einer gespeicherten Prozedur oder Funktion verwenden möchten. AWS SCT emulierte diesen Typ in PostgreSQL, indem eine temporäre Tabelle erstellt wurde.

AWS SCT Konvertiert bei der Konvertierung von SQL Server nach PostgreSQL SQL Server-Systemobjekte in erkennbare Objekte in PostgreSQL. Die folgende Tabelle zeigt, wie die Systemobjekte umgewandelt werden.

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
SYS.SCHEMAS	AWS_SQLSERVER_EXT.SYS_SCHEMAS
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABELLEN
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_ANSICHTEN
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALLE_ANSICHTEN
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPEN
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_SPALTEN
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALLE_SPALTEN
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FREMDSCHLÜSSEL
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYS FREMDSCHLÜSSEL
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_PROZEDUREN

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDIZES
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXE
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALLE_OBJEKTE
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJEKTE
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULE
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATENBANKEN
INFORMATION_SCHEMA.SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA.VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ANSICHTEN
INFORMATION_SCHEMA.TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABELLEN
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

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
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_ROUTINEN
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROZESSE
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

Verwenden eines AWS SCT Erweiterungspakets zum Emulieren des SQL Server-Agents in PostgreSQL

SQL Server Agent ist ein Microsoft Windows-Dienst, der SQL Server-Jobs ausführt. Der SQL Server-Agent führt Jobs nach einem Zeitplan, als Reaktion auf ein bestimmtes Ereignis oder bei Bedarf aus. Weitere Informationen zum SQL Server-Agent finden Sie in der technischen Dokumentation von Microsoft.

PostgreSQL hat kein Äquivalent für SQL Server Agent. AWS SCT Erstellt ein Erweiterungspaket, um die Funktionen des SQL Server-Agents zu emulieren. Dieses Erweiterungspaket verwendet AWS Lambda Amazon CloudWatch. AWS Lambda implementiert die Schnittstelle, die Sie zum Verwalten von Zeitplänen und zum Ausführen von Jobs verwenden. Amazon CloudWatch hält die Zeitplanregeln ein.

AWS Lambda und Amazon CloudWatch verwendet einen JSON-Parameter für die Interaktion. Dieser JSON-Parameter hat die folgende Struktur.

```
{
```

```
    "mode": mode,
    "parameters": {
        list of parameters
    },
    "callback": procedure name
}
```

Im vorherigen Beispiel *mode* ist dies der Typ der Aufgabe und *list of parameters* besteht aus einer Reihe von Parametern, die vom Typ der Aufgabe abhängen. Dies *procedure name* ist auch der Name der Prozedur, die nach Abschluss der Aufgabe ausgeführt wird.

AWS SCT verwendet eine Lambda-Funktion zur Steuerung und Ausführung von Jobs. Die CloudWatch Regel startet die Ausführung des Jobs und stellt die erforderlichen Informationen zum Starten des Jobs bereit. Wenn die CloudWatch Regel ausgelöst wird, startet sie die Lambda-Funktion mit den Parametern aus der Regel.

Verwenden Sie das folgende Format, um einen einfachen Job zu erstellen, der eine Prozedur aufruft.

```
{
    "mode": "run_job",
    "parameters": {
        "vendor": "mysql",
        "cmd": "lambda_db.nightly_job"
    }
}
```

Verwenden Sie das folgende Format, um einen Job mit mehreren Schritten zu erstellen.

```
{
    "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
        },
        ...
]
}
```

Um das Verhalten des SQL Server-Agents in PostgreSQL zu emulieren, erstellt das AWS SCT Erweiterungspaket auch die folgenden Tabellen und Prozeduren.

Tabellen, die den SQL Server-Agent in PostgreSQL emulieren

Um den SQL Server-Agent zu emulieren, verwendet das Erweiterungspaket die folgenden Tabellen:

sysjobs

Speichert die Informationen über die Jobs.

sysjobsteps

Speichert die Informationen über die Schritte eines Jobs.

Sysschedules

Speichert die Informationen über die Jobpläne.

sysjobschedules

Speichert die Zeitplaninformationen für einzelne Jobs.

sysjobhistory

Speichert die Informationen über die Ausführung von geplanten Aufträgen.

Verfahren, die den SQL Server-Agent in PostgreSQL emulieren

Um den SQL Server-Agent zu emulieren, verwendet das Erweiterungspaket die folgenden Verfahren:

sp_add_job

Fügt einen neuen Job hinzu.

sp_add_jobstep

Fügt einem Job einen Schritt hinzu.

sp_add_schedule

Erstellt eine neue Zeitplanregel in Amazon CloudWatch. Sie können diesen Zeitplan für eine beliebige Anzahl von Jobs verwenden.

sp_attach_schedule

Legt einen Zeitplan für den ausgewählten Job fest.

sp_add_jobschedule

Erstellt eine Zeitplanregel für einen Job in Amazon CloudWatch und legt das Ziel für diese Regel fest.

sp_update_job

Aktualisiert die Attribute des zuvor erstellten Jobs.

sp_update_jobstep

Aktualisiert die Attribute des Schritts in einem Job.

sp_update_schedule

Aktualisiert die Attribute einer Zeitplanregel in Amazon CloudWatch.

sp_update_jobschedule

Aktualisiert die Attribute des Zeitplans für den angegebenen Job.

sp_delete_job

Löscht einen Job.

sp_delete_jobstep

Löscht einen Job-Schritt aus einem Job.

sp_delete_schedule

Löscht einen Zeitplan.

sp_delete_jobschedule

Löscht die Zeitplanregel für den angegebenen Job von Amazon CloudWatch.

sp_detach_schedule

Entfernt eine Verknüpfung zwischen einem Zeitplan und einem Job.

get_jobs, update_job

Interne Verfahren, die interagieren mit. 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

Interne Verfahren, die Einstellungen überprüfen.

Syntax für Prozeduren, die den SQL Server-Agent in PostgreSQL emulieren

Die `aws_sqlserver_ext.sp_add_job` Prozedur im Erweiterungspaket emuliert die Prozedur. `msdb.dbo.sp_add_job` Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_add_jobstep` Prozedur emuliert die `msdb.dbo.sp_add_jobstep` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_add_schedule` Prozedur emuliert die `msdb.dbo.sp_add_schedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_attach_schedule` Prozedur emuliert die `msdb.dbo.sp_attach_schedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_add_jobschedule` Prozedur emuliert die `msdb.dbo.sp_add_jobschedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_delete_job` Prozedur emuliert die `msdb.dbo.sp_delete_job` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_delete_jobstep` Prozedur emuliert die `msdb.dbo.sp_delete_jobstep` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von Microsoft.

```
par_job_id integer = NULL::integer,
par_job_name varchar = NULL::character varying,
par_step_id integer = NULL::integer,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_delete_jobschedule` Prozedur emuliert die `msdb.dbo.sp_delete_jobschedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_delete_schedule` Prozedur emuliert die `msdb.dbo.sp_delete_schedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_detach_schedule` Prozedur emuliert die `msdb.dbo.sp_detach_schedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_update_job` Prozedur emuliert die `msdb.dbo.sp_update_job` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_update_jobschedule` Prozedur emuliert die `msdb.dbo.sp_update_jobschedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_update_jobstep` Prozedur emuliert die `msdb.dbo.sp_update_jobstep` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sp_update_schedule` Prozedur emuliert die `msdb.dbo.sp_update_schedule` Prozedur. Weitere Informationen zur Quellprozedur des SQL Server-Agents finden Sie in der technischen Dokumentation von 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
```

Beispiele für die Verwendung von Prozeduren, die den SQL Server-Agent in PostgreSQL emulieren

Gehen Sie wie `aws_sqlserver_ext.sp_add_job` folgt vor, um einen neuen Job hinzuzufügen.

```
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');
```

Gehen Sie wie `aws_sqlserver_ext.sp_add_jobstep` folgt vor, um einen neuen Auftragsschritt hinzuzufügen.

```
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');
```

Gehen Sie wie `aws_sqlserver_ext.sp_add_schedule` folgt vor, um einen einfachen Zeitplan hinzuzufügen.

```
SELECT * FROM aws_sqlserver_ext.sp_add_schedule(
    par_schedule_name := 'RunOnce',
    par_freq_type := 1,
    par_active_start_time := 233000);
```

`aws_sqlserver_ext.sp_attach_schedule`Gehen Sie wie folgt vor, um einen Zeitplan für einen Job festzulegen.

```
SELECT * FROM aws_sqlserver_ext.sp_attach_schedule (
    par_job_name := 'test_job',
    par_schedule_name := 'NightlyJobs');
```

`aws_sqlserver_ext.sp_add_jobschedule`Gehen Sie wie folgt vor, um einen Zeitplan für einen Job zu erstellen.

```
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);
```

Anwendungsbeispiele für die Emulation des SQL Server-Agents in PostgreSQL

Wenn Ihr Quelldatenbankcode den SQL Server-Agent zum Ausführen von Aufträgen verwendet, können Sie das SQL Server to PostgreSQL-Erweiterungspaket für verwenden, um diesen Code AWS SCT nach PostgreSQL zu konvertieren. Das Erweiterungspaket verwendet AWS Lambda Funktionen, um das Verhalten des SQL Server-Agents zu emulieren.

Sie können eine neue AWS Lambda Funktion erstellen oder eine vorhandene Funktion registrieren.

Um eine neue AWS Lambda Funktion zu erstellen

1. Öffnen Sie in der Zieldatenbankstruktur das Kontextmenü (Rechtsklick), wählen Sie Erweiterungspaket anwenden für und wählen Sie dann PostgreSQL. AWS SCT

   Der Assistent des Erweiterungspakets wird angezeigt.
2. Gehen Sie auf der Registerkarte SQL Server-Agent-Emulationsdienst wie folgt vor:

   - Wählen Sie AWS Lambda Funktion erstellen aus.
   - Geben Sie für Datenbank-Login den Namen des Zieldatenbankbenutzers ein.
   - Geben Sie unter Datenbankkennwort das Passwort für den Benutzernamen ein, den Sie im vorherigen Schritt eingegeben haben.
   - Geben Sie für Python-Bibliotheksordner den Pfad zu Ihrem Python-Bibliotheksordner ein.
   - Wählen Sie „ AWS Lambda Funktion erstellen“ und dann „Weiter“.

Um eine AWS Lambda Funktion zu registrieren, die Sie zuvor bereitgestellt haben

- Führen Sie das folgende Skript in Ihrer Zieldatenbank aus.

```
SELECT
    FROM aws_sqlserver_ext.set_service_setting(
```

```
        p_service := 'JOB',
        p_setting := 'LAMBDA_ARN',
        p_value := ARN)
```

Im vorherigen Beispiel *ARN* ist dies der Amazon-Ressourcenname (ARN) der bereitgestellten AWS Lambda Funktion.

Im folgenden Beispiel wird eine einfache Aufgabe erstellt, die aus einem Schritt besteht. Diese Aufgabe führt alle fünf Minuten die zuvor erstellte `job_example` Funktion aus. Diese Funktion fügt Datensätze in die `job_example_table` Tabelle ein.

Um diese einfache Aufgabe zu erstellen

1. Erstellen Sie einen Job mit der `aws_sqlserver_ext.sp_add_job` Funktion wie im Folgenden gezeigt.

```
SELECT
    FROM aws_sqlserver_ext.sp_add_job (
        par_job_name := 'test_simple_job');
```

2. Erstellen Sie einen Auftragsschritt mithilfe der `aws_sqlserver_ext.sp_add_jobstep` Funktion, wie im Folgenden gezeigt.

```
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;');
```

   Der Job-Schritt gibt an, was die Funktion macht.

3. Erstellen Sie mithilfe der `aws_sqlserver_ext.sp_add_jobschedule` Funktion einen Scheduler für den Job, wie im Folgenden gezeigt.

```
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 */);
```

Der Job-Schritt gibt an, was die Funktion macht.

Um diesen Job zu löschen, verwenden Sie die `aws_sqlserver_ext.sp_delete_job` Funktion wie folgt.

```
PERFORM aws_sqlserver_ext.sp_delete_job(
    par_job_name := 'PeriodicJob1'::character varying,
    par_delete_history := 1::smallint,
    par_delete_unused_schedule := 1::smallint);
```

Verwenden eines AWS SCT Erweiterungspakets zum Emulieren von SQL Server-Datenbank-Mail in PostgreSQL

Sie können SQL Server Database Mail verwenden, um E-Mail-Nachrichten von der SQL Server Database Engine oder Azure SQL Managed Instance aus an Benutzer zu senden. Diese E-Mail-Nachrichten können Abfrageergebnisse oder Dateien aus beliebigen Ressourcen in Ihrem Netzwerk enthalten. Weitere Informationen zu SQL Server Database Mail finden Sie in der technischen Dokumentation von Microsoft.

PostgreSQL hat kein Äquivalent für SQL Server Database Mail. AWS SCT Erstellt ein Erweiterungspaket, um die Funktionen von SQL Server Database Mail zu emulieren. Dieses Erweiterungspaket verwendet AWS Lambda einen Amazon Simple Email Service (Amazon SES). AWS Lambda bietet Benutzern eine Schnittstelle zur Interaktion mit dem Amazon SES SES-E-Mail-Versandservice. Um diese Interaktion einzurichten, fügen Sie den Amazon-Ressourcennamen (ARN) Ihrer Lambda-Funktion hinzu.

Verwenden Sie für ein neues E-Mail-Konto den folgenden Befehl.

```
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;
```

Verwenden Sie den folgenden Befehl, um den ARN Ihrer Lambda-Funktion zum vorhandenen E-Mail-Konto hinzuzufügen.

```
do
$$
begin
PERFORM sysmail_update_account_sp (
    par_account_name :='existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

In den vorherigen Beispielen *ARN* ist das der ARN Ihrer Lambda-Funktion.

Um das Verhalten von SQL Server-Datenbankmail in PostgreSQL zu emulieren, verwendet das AWS SCT Erweiterungspaket die folgenden Tabellen, Ansichten und Prozeduren.

Tabellen, die SQL Server-Datenbankmail in PostgreSQL emulieren

Um SQL Server Database Mail zu emulieren, verwendet das Erweiterungspaket die folgenden Tabellen:

sysmail_account

Speichert die Informationen über die E-Mail-Konten.

sysmail_profile

Speichert die Informationen zu den Benutzerprofilen.

sysmail_server

Speichert die Informationen über die E-Mail-Server.

sysmail_mailitems

Speichert die Liste der E-Mail-Nachrichten.

sysmail_attachments

Enthält eine Zeile für jeden E-Mail-Anhang.

sysmail_log

Speichert die Dienstinformationen zum Senden von E-Mail-Nachrichten.

sysmail_profile_account

Speichert die Informationen zu den Benutzerprofilen und E-Mail-Konten.

Ansichten, die SQL Server-Datenbankmail in PostgreSQL emulieren

Um SQL Server Database Mail zu emulieren, AWS SCT erstellt die folgenden Ansichten in der PostgreSQL-Datenbank, um die Kompatibilität sicherzustellen. Das Erweiterungspaket verwendet sie nicht, aber Ihr konvertierter Code kann diese Ansichten abfragen.

sysmail_allitems

Enthält eine Liste aller E-Mails.

sysmail_faileditems

Enthält eine Liste von E-Mails, die nicht gesendet werden konnten.

sysmail_sentitems

Enthält eine Liste der gesendeten E-Mails.

sysmail_unsentitems

Enthält eine Liste von E-Mails, die noch nicht gesendet wurden.

sysmail_mailattachments

Enthält eine Liste der angehängten Dateien.

Verfahren, die SQL Server-Datenbank-Mail in PostgreSQL emulieren

Um SQL Server Database Mail zu emulieren, verwendet das Erweiterungspaket die folgenden Verfahren:

sp_send_dbmail

Sendet eine E-Mail an die angegebenen Empfänger.

sysmail_add_profile_sp

Erstellt ein neues Benutzerprofil.

sysmail_add_account_sp

Erstellt ein neues E-Mail-Konto, in dem Informationen wie SMTP-Anmeldeinformationen (Simple Mail Transfer Protocol) usw. gespeichert werden.

sysmail_add_profileaccount_sp

Fügt dem angegebenen Benutzerprofil ein E-Mail-Konto hinzu.

sysmail_update_profile_sp

Ändert die Attribute des Benutzerprofils wie Beschreibung, Name usw.

sysmail_update_account_sp

Ändert die Informationen im vorhandenen E-Mail-Konto.

sysmail_update_profileaccount_sp

Aktualisiert die E-Mail-Kontoinformationen im angegebenen Benutzerprofil.

sysmail_delete_profileaccount_sp

Entfernt ein E-Mail-Konto aus dem angegebenen Benutzerprofil.

sysmail_delete_account_sp

Löscht das E-Mail-Konto.

sysmail_delete_profile_sp

Löscht das Benutzerprofil.

sysmail_delete_mailitems_sp

Löscht E-Mails aus internen Tabellen.

sysmail_help_profile_sp

Zeigt Informationen über das Benutzerprofil an.

sysmail_help_account_sp

Zeigt Informationen über das E-Mail-Konto an.

sysmail_help_profileaccount_sp

Zeigt Informationen zu E-Mail-Konten an, die dem Benutzerprofil zugeordnet sind.

sysmail_dbmail_json

Eine interne Prozedur, die JSON-Anfragen für Funktionen generiert. AWS Lambda

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Interne Verfahren, die Einstellungen überprüfen.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Veraltete interne Prozeduren.

Syntax für Prozeduren, die SQL Server-Datenbankmail in PostgreSQL emulieren

Die `aws_sqlserver_ext.sp_send_dbmail` Prozedur im Erweiterungspaket emuliert die Prozedur. `msdb.dbo.sp_send_dbmail` Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_delete_mailitems_sp` Prozedur emuliert die `msdb.dbo.sysmail_delete_mailitems_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_sent_before timestamp = NULL::timestamp without time zone,
par_sent_status varchar = NULL::character varying,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_add_profile_sp` Prozedur emuliert die `msdb.dbo.sysmail_add_profile_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_profile_name varchar,
par_description varchar = NULL::character varying,
out par_profile_id integer,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_add_account_sp` Prozedur emuliert die `msdb.dbo.sysmail_add_account_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_add_profileaccount_sp` Prozedur emuliert die `msdb.dbo.sysmail_add_profileaccount_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_help_profile_sp` Prozedur emuliert die `msdb.dbo.sysmail_help_profile_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_update_profile_sp` Prozedur emuliert die `msdb.dbo.sysmail_update_profile_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
par_description varchar = NULL::character varying,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_delete_profile_sp` Prozedur emuliert die `msdb.dbo.sysmail_delete_profile_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_profile_id integer = NULL::integer,
```

```
par_profile_name varchar = NULL::character varying,
par_force_delete smallint = 1,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_help_account_sp` Prozedur emuliert die `msdb.dbo.sysmail_help_account_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_account_id integer = NULL::integer,
par_account_name varchar = NULL::character varying,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_update_account_sp` Prozedur emuliert die `msdb.dbo.sysmail_update_account_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_delete_account_sp` Prozedur emuliert die `msdb.dbo.sysmail_delete_account_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von Microsoft.

```
par_account_id integer = NULL::integer,
```

```
par_account_name varchar = NULL::character varying,
out returncode integer
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_help_profileaccount_sp` Prozedur emuliert die `msdb.dbo.sysmail_help_profileaccount_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_update_profileaccount_sp` Prozedur emuliert die `msdb.dbo.sysmail_update_profileaccount_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Die im Erweiterungspaket enthaltene `aws_sqlserver_ext.sysmail_delete_profileaccount_sp` Prozedur emuliert die `msdb.dbo.sysmail_delete_profileaccount_sp` Prozedur. Weitere Informationen zum Quellverfahren für SQL Server-Datenbankmail finden Sie in der technischen Dokumentation von 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
```

Beispiele für die Verwendung von Prozeduren, die SQL Server-Datenbankmail emulieren, in PostgreSQL

Gehen Sie wie `aws_sqlserver_ext.sp_send_dbmail` folgt vor, um eine E-Mail zu senden.

```
PERFORM sp_send_dbmail (
    par_profile_name := 'Administrator',
    par_recipients := 'hello@rusgl.info',
    par_subject := 'Automated Success Message',
    par_body := 'The stored procedure finished'
);
```

Das folgende Beispiel zeigt, wie eine E-Mail mit Abfrageergebnissen gesendet wird.

```
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'
);
```

Das folgende Beispiel zeigt, wie eine E-Mail mit HTML-Code gesendet wird.

```
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'
);
```

Gehen Sie wie `aws_sqlserver_ext.sysmail_delete_mailitems_sp` folgt vor, um E-Mails zu löschen.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := var_GETDATE
);
```

Das folgende Beispiel zeigt, wie die ältesten E-Mails gelöscht werden.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := '31.12.2015'
);
```

Das folgende Beispiel zeigt, wie Sie alle E-Mails löschen, die nicht gesendet werden können.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_status := 'failed'
);
```

Gehen Sie wie `aws_sqlserver_ext.sysmail_add_profile_sp` folgt vor, um ein neues Benutzerprofil zu erstellen.

```
PERFORM sysmail_add_profile_sp (
    profile_name := 'Administrator',
    par_description := 'administrative mail'
);
```

Das folgende Beispiel zeigt, wie Sie ein neues Profil erstellen und die eindeutige Profil-ID in einer Variablen speichern.

```
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;
```

Gehen Sie wie `aws_sqlserver_ext.sysmail_add_account_sp` folgt vor, um ein neues E-Mail-Konto zu erstellen.

```
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'
);
```

`aws_sqlserver_ext.sysmail_add_profileaccount_sp`Gehen Sie wie folgt vor, um dem Benutzerprofil ein E-Mail-Konto hinzuzufügen.

```
PERFORM sysmail_add_profileaccount_sp (
    par_account_name := 'Administrator',
    par_account_name := 'Audit Account',
    par_sequence_number := 1
);
```

Anwendungsbeispiele für die Emulation von SQL Server-Datenbank-Mail in PostgreSQL

Wenn Ihr Quelldatenbankcode SQL Server Database Mail zum Senden von E-Mails verwendet, können Sie das AWS SCT Erweiterungspaket verwenden, um diesen Code nach PostgreSQL zu konvertieren.

Um eine E-Mail aus Ihrer PostgreSQL-Datenbank zu senden

1. Erstellen und konfigurieren Sie Ihre AWS Lambda Funktion.
2. Wenden Sie das AWS SCT Erweiterungspaket an.
3. Erstellen Sie ein Benutzerprofil mit der `sysmail_add_profile_sp` Funktion, wie im Folgenden gezeigt.
4. Erstellen Sie mit der folgenden `sysmail_add_account_sp` Funktion ein E-Mail-Konto.
5. Fügen Sie dieses E-Mail-Konto mithilfe der folgenden `sysmail_add_profileaccount_sp` Funktion zu Ihrem Benutzerprofil hinzu.

   ```
   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. Senden Sie eine E-Mail mit der `sp_send_dbmail` Funktion wie im Folgenden gezeigt.

```
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;
```

`sysmail_help_profile_sp`Gehen Sie wie folgt vor, um die Informationen zu allen Benutzerprofilen anzuzeigen.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

Im folgenden Beispiel werden die Informationen über das spezifische Benutzerprofil angezeigt.

```
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');
```

`sysmail_help_account_sp`Gehen Sie wie folgt vor, um die Informationen zu allen E-Mail-Konten anzuzeigen.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

Im folgenden Beispiel werden die Informationen zu dem jeweiligen E-Mail-Konto angezeigt.

```
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');
```

`sysmail_help_profileaccount_sp`Gehen Sie wie folgt vor, um die Informationen zu allen E-Mail-Konten anzuzeigen, die mit den Benutzerprofilen verknüpft sind.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

Im folgenden Beispiel werden die Datensätze nach ID, Profilname oder Kontoname gefiltert.

```
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');
```

`sysmail_update_profile_sp`Gehen Sie wie folgt vor, um den Namen oder die Beschreibung des Benutzerprofils zu ändern.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
    par_profile_id := 2,
    par_profile_name := 'New profile name'
```

```
);
```

Gehen Sie wie `ysmail_update_account_sp` folgt vor, um die E-Mail-Kontoeinstellungen zu ändern.

```
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 von SQL Server zu Amazon RDS for SQL Server mit AWS Schema Conversion Tool

Bei der Migration von SQL Server-Schema und -Code in Amazon RDS für SQL Server sind einige Punkte zu beachten:

- AWS SCT kann den SQL Server-Agent konvertieren, um Zeitpläne, Benachrichtigungen und Jobs auf einer Amazon RDS for SQL Server-DB-Instance bereitzustellen. Nach der Umwandlung können Sie eine Amazon RDS für SQL Server-DB-Instance mit SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS) und SQL Server Integration Services (SSIS) verwenden.
- Amazon RDS unterstützt derzeit SQL Server Service Broker nicht, und auch keine zusätzlichen T-SQL-Endpunkte, für die Sie den Befehl CREATE ENDPOINT ausführen müssen.
- Amazon RDS verfügt über begrenzten Support für verknüpfte Server. Bei der Konvertierung von SQL Server-Anwendungscode, der verknüpfte Server verwendet, AWS SCT wird der Anwendungscode konvertiert. Überprüfen Sie jedoch das Verhalten von Objekten, die Verbindungsserver verwenden, bevor Sie den umgewandelten Code ausführen.
- AlwaysOn wird verwendet.
- Der AWS SCT Bewertungsbericht enthält Servermetriken für die Konvertierung. Zu diesen Metriken über Ihre SQL Server-Instance gehören:
  - Datenspiegelung wird verwendet.
  - SQL Server-Protokollversand ist konfiguriert.
  - Failover-Cluster wird verwendet.
  - Datenbank-E-Mail ist konfiguriert.
  - Full Text Search Service wird verwendet. Amazon RDS für SQL Server verfügt über eine begrenzte Volltextsuche und unterstützt keine semantische Suche.

- Data Quality Service (DQS) ist installiert. Amazon RDS unterstützt DQS nicht, daher empfehlen wir, SQL Server auf einer EC2 Amazon-Instance zu installieren.

Rechte für RDS für SQL Server als Ziel

Um zu RDS für SQL Server zu migrieren, erstellen Sie einen Datenbankbenutzer und gewähren Sie dann die erforderlichen Rechte für jede Datenbank. Sie können das folgende Codebeispiel verwenden.

```
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;
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *db_name* durch den Namen Ihrer Zieldatenbank. Schließlich ersetzen Sie es *your_password* durch ein sicheres Passwort.

Data Warehouse-Quellen für AWS Schema Conversion Tool

AWS SCT kann Schemas für die folgenden Quell-Data Warehouses in ein unterstütztes Ziel konvertieren. Informationen zu Berechtigungen, Verbindungen und Informationen darüber, was für die Verwendung mit der Zieldatenbank oder dem Data Warehouse konvertiert werden AWS SCT kann, finden Sie im Folgenden.

Themen

- Verbinden von Amazon Redshift mit dem AWS Schema Conversion Tool
- Azure Synapse Analytics verbinden mit AWS Schema Conversion Tool
- Verbindung zu Google herstellen BigQuery mit AWS Schema Conversion Tool
- Greenplum Database verbinden mit AWS Schema Conversion Tool
- Verbindung zu Netezza herstellen mit AWS Schema Conversion Tool
- Oracle Data Warehouse verbinden mit AWS SCT
- Verbindung zu einem Snowflake-Data Warehouse herstellen mit AWS Schema Conversion Tool
- Herstellen einer Verbindung zu einem SQL Server Data Warehouse mit dem AWS Schema Conversion Tool
- Herstellen einer Verbindung zu einem Teradata Data Warehouse mit dem AWS Schema Conversion Tool
- Verbindung der mit den AWS Schema Conversion Tool Vertica-Datenbanken herstellen

Verbinden von Amazon Redshift mit dem AWS Schema Conversion Tool

Sie können es verwenden AWS SCT , um Ihren Amazon Redshift Redshift-Cluster zu optimieren. AWS SCT gibt Ihnen Empfehlungen zur Auswahl von Verteilungs- und Sortierschlüsseln für Ihren Amazon Redshift Redshift-Cluster. Sie können das Amazon Redshift Redshift-Optimierungsprojekt als ein AWS SCT Projekt betrachten, bei dem Quelle und Ziel auf die verschiedenen Amazon Redshift Redshift-Cluster verweisen.

Rechte für Amazon Redshift als Quelldatenbank

Für die Verwendung von Amazon Redshift als Quelle sind die folgenden Rechte erforderlich:

- VERWENDUNG IM SCHEMA *<schema_name>*
- WÄHLEN SIE FÜR ALLE TABELLEN IM SCHEMA *<schema_name>*
- SELECT ON PG_CATALOG.PG_STATISTIC
- SELECT ON SVV_TABLE_INFO
- SELECT ON TABLE STV_BLOCKLIST
- SELECT ON TABLE STV_TBL_PERM
- WÄHLEN SIE FÜR SYS_SERVERLESS_USAGE
- WÄHLEN SIE AUF PG_DATABASE_INFO
- WÄHLEN SIE AUF PG_STATISTIC

Ersetzen Sie in den vorherigen Beispielen den *<schema_name>* Platzhalter durch den Namen des Quellschemas.

Informationen zu den für Amazon Redshift als Ziel erforderlichen Rechten finden Sie unterBerechtigungen für Amazon Redshift als Ziel.

Verbindung zu Amazon Redshift als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Amazon Redshift Redshift-Quelldatenbank herzustellen. AWS Schema Conversion Tool

So stellen Sie eine Verbindung zu einer Amazon Redshift Redshift-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen.
2. Wählen Sie Amazon Redshift und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen für die Amazon Redshift Redshift-Quelldatenbank einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.

Parameter	Aktion
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Amazon-Redshift-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Vertrauensspeichers zu überprüfen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Speicherort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen. Weitere Informationen zur SSL-Unterstützung für Amazon Redshift finden Sie unter Sicherheitsoptionen für Verbindungen konfigurieren.

Parameter	Aktion
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Redshift-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Amazon Redshift Redshift-Optimierungseinstellungen

Um die Amazon Redshift Redshift-Optimierungseinstellungen zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Amazon Redshift und dann Amazon Redshift — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Amazon Redshift Redshift-Optimierung an.

Die Amazon Redshift Redshift-Optimierungseinstellungen AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus.

AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn die Anzahl der Tabellen größer ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Migrationsstrategie auszuwählen.

  AWS empfiehlt, verschiedene Cluster als Quelle und Ziel für Ihr Optimierungsprojekt zu verwenden. Vor dem Start des Amazon Redshift Redshift-Optimierungsprozesses erstellen Sie eine Kopie Ihres Amazon Redshift Redshift-Quellclusters. Sie können Ihre Quelldaten in diese Kopie aufnehmen oder einen leeren Cluster erstellen.

  Wählen Sie unter Migrationsstrategie die Option Migration zu einer Kopie aus, um Daten aus Ihrem Quellcluster in den Zielcluster aufzunehmen.

  Wählen Sie unter Migrationsstrategie die Option Migration to a Clean Slate aus, um die Optimierungsvorschläge zu überprüfen. Nachdem Sie diese Vorschläge akzeptiert haben, migrieren Sie Ihre Quelldaten in den Zielcluster.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden ausgewählt haben.
- Um mit der automatischen Tabellenoptimierung zu arbeiten.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Um diese Option nur für die automatische Tabellenoptimierung zu verwenden, wählen Sie Optimierungsstrategien im linken Bereich aus. Wählen Sie dann Automatische Tabellenoptimierung von Amazon Redshift verwenden und wählen Sie Keine für Strategie zur anfänglichen Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignoriere Metadaten, verwende statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
  - Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, deren Schrägwert über dem Schwellenwert liegt, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
  - Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
  - Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, dessen Abfragestatistiken Sie analysieren möchten.

Azure Synapse Analytics verbinden mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Azure Synapse Analytics nach Amazon Redshift zu konvertieren.

Rechte für Azure Synapse Analytics als Quelldatenbank

Die folgenden Rechte sind erforderlich, um ein Azure Synapse Analytics-Data Warehouse als Quelle zu verwenden:

- VIEW DEFINITION
- VIEW DATABASE STATE

Wenden Sie die Berechtigungen für jede Datenbank an, deren Schema Sie konvertieren.

Verbindung zu Azure Synapse Analytics als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrem Azure Synapse Analytics-Data Warehouse herzustellen. AWS Schema Conversion Tool

Um eine Verbindung zu einem Azure Synapse Analytics-Data Warehouse als Quelle herzustellen

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen aus.
2. Wählen Sie Azure Synapse Analytics und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen für das Azure Synapse Analytics-Data Warehouse manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
SQL-Pool	Geben Sie den Namen des Azure SQL-Pools ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie

Parameter	Aktion
	sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat vertrauen: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen. • Vertrauensspeicher: Ein Vertrauensspeicher, den Sie in den globalen Einstellungen eingerichtet haben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Azure Synapse Analytics nach Amazon Redshift

Um die Konvertierungseinstellungen von Azure Synapse Analytics zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Azure Synapse und dann Azure Synapse — Amazon Redshift aus.

AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Azure Synapse Analytics zu Amazon Redshift an.

Die Konvertierungseinstellungen von Azure Synapse Analytics zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.
- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Einstellungen für die Optimierung der Konvertierung von Azure Synapse Analytics zu Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Azure Synapse Analytics zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Azure Synapse und dann Azure Synapse — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Azure Synapse Analytics zu Amazon Redshift an.

Die Einstellungen für die Optimierung der Konvertierung von Azure Synapse Analytics zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignoriere Metadaten, verwende statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.
- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.

- Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verbindung zu Google herstellen BigQuery mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von BigQuery zu Amazon Redshift zu konvertieren.

Rechte für BigQuery als Quelle

Um ein BigQuery Data Warehouse als Quelle in zu verwenden AWS SCT, erstellen Sie ein Dienstkonto. In Google Cloud verwenden Anwendungen Dienstkonten, um autorisierte API-Aufrufe zu tätigen. Dienstkonten unterscheiden sich von Benutzerkonten. Weitere Informationen finden Sie unter Dienstkonten in der Google Cloud Identity and Access Management-Dokumentation.

Stellen Sie sicher, dass Sie Ihrem Dienstkonto die folgenden Rollen zuweisen:

- `BigQuery Admin`
- `Storage Admin`

Die `BigQuery Admin` Rolle bietet Berechtigungen zur Verwaltung aller Ressourcen innerhalb des Projekts. AWS SCT verwendet diese Rolle, um Ihre BigQuery Metadaten in das Migrationsprojekt zu laden.

Die `Storage Admin` Rolle gewährt die volle Kontrolle über Datenobjekte und Buckets. Sie finden diese Rolle unter`Cloud Storage`. AWS SCT verwendet diese Rolle, um Ihre Daten aus Amazon Redshift zu extrahieren BigQuery und sie dann in Amazon Redshift zu laden.

Um eine Dienstkonto-Schlüsseldatei zu erstellen

1. Melden Sie sich unter https://console.cloud.google.com/bei der Google Cloud-Verwaltungskonsole an.
2. Wählen Sie auf der BigQuery API-Seite die Option Aktivieren aus. Überspringen Sie diesen Schritt, wenn Sie „API aktiviert“ sehen.
3. Wählen Sie auf der Seite Dienstkonten Ihr Projekt aus und klicken Sie dann auf Dienstkonto erstellen.
4. Geben Sie auf der Seite mit den Dienstkontodetails einen beschreibenden Wert für den Namen des Dienstkontos ein. Wählen Sie Erstellen und fortfahren. Die Seite „Diesem Dienstkonto Zugriff auf das Projekt gewähren“ wird geöffnet.
5. Wählen Sie unter Rolle auswählen BigQuerydie Option und anschließend BigQuery Admin aus.
6. Wählen Sie Weitere Rolle hinzufügen aus. Wählen Sie für Rolle auswählen die Option Cloud Storage und dann Storage Admin aus.
7. Wählen Sie Weiter und dann Fertig.
8. Wählen Sie auf der Seite Dienstkonten das Dienstkonto aus, das Sie erstellt haben.
9. Wählen Sie Schlüssel und dann unter Schlüssel hinzufügen die Option Neuen Schlüssel erstellen aus.
10. Wählen Sie JSON und dann Create aus. Wählen Sie den Ordner aus, in dem Ihr privater Schlüssel gespeichert werden soll, oder wählen Sie den Standardordner für Downloads in Ihrem Browser aus.

Zum Extrahieren von Daten aus einem BigQuery Data Warehouse AWS SCT wird der Bucket-Ordner von Google Cloud Storage verwendet. Erstellen Sie diesen Bucket, bevor Sie mit der Datenmigration beginnen. Geben Sie im Dialogfeld „Lokale Aufgabe erstellen“ den Pfad zu Ihrem Google Cloud Storage-Bucket-Ordner ein. Weitere Informationen finden Sie unter Eine AWS SCT Aufgabe erstellen, ausführen und überwachen.

Verbindung zu BigQuery als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrem BigQuery Quellprojekt herzustellen AWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einem BigQuery Quell-Data Warehouse her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen aus.
2. Wählen BigQuerySie und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihr BigQuery Projekt ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Geben Sie unter Schlüsselpfad den Pfad zur Schlüsseldatei des Dienstkontos ein. Weitere Informationen zum Erstellen dieser Datei finden Sie unterRechte für BigQuery als Quelle.
5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrem BigQuery Quellprojekt hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrem BigQuery Quellprojekt herzustellen.

Einschränkungen bei der Verwendung BigQuery als Quelle für AWS SCT

Bei der Verwendung BigQuery als Quelle für gelten die folgenden Einschränkungen AWS SCT:

- AWS SCT unterstützt die Konvertierung von Unterabfragen in Analysefunktionen nicht.
- Sie können nicht AWS SCT zur Konvertierung BigQuery `SELECT AS STRUCT` von AND-Anweisungen verwenden. `SELECT AS VALUE`
- AWS SCT unterstützt die Konvertierung der folgenden Funktionstypen nicht:
  - Ungefähre Summe
  - Bit
  - Debugging
  - Föderierte Abfrage
  - Geografie
  - Hash
  - Mathematisch
  - Netz
  - Statistisches Aggregat
  - UUID
- AWS SCT bietet eingeschränkte Unterstützung für die Konvertierung von Zeichenkettenfunktionen.
- AWS SCT unterstützt die Konvertierung von `UNNEST` Operatoren nicht.

- Sie können korrelierte Join-Operationen nicht in AWS SCT konvertieren.
- AWS SCT unterstützt die Konvertierung von`QUALIFY`,`WINDOW`, und `LIMIT OFFSET` -Klauseln nicht.
- Sie können es nicht verwenden AWS SCT , um rekursive allgemeine Tabellenausdrücke zu konvertieren.
- AWS SCT unterstützt nicht die Konvertierung von `INSERT` Anweisungen mit Unterabfragen innerhalb `VALUES` von Klauseln.
- AWS SCT unterstützt nicht die Konvertierung von `UPDATE` Anweisungen für verschachtelte Felder und wiederholte Datensätze.
- Sie können die Konvertierung von `STRUCT ARRAY` Datentypen nicht verwenden AWS SCT .

BigQuery zu Amazon Redshift Redshift-Konvertierungseinstellungen

Um die Konvertierungseinstellungen BigQuery zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Google BigQuery und dann Google BigQuery — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung BigQuery zu Amazon Redshift an.

BigQuery zu Amazon Redshift Redshift-Konvertierungseinstellungen in AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

BigQuery zu den Einstellungen für die Konvertierungsoptimierung von Amazon Redshift

Um die Einstellungen BigQuery für die Konversionsoptimierung in Amazon Redshift zu ändern, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Google BigQuery und dann Google BigQuery — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die BigQuery Konvertierung in Amazon Redshift an.

BigQuery zu den Einstellungen für die Amazon Redshift Redshift-Konversionsoptimierung, AWS SCT einschließlich Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignoriere Metadaten, verwende statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.
- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.

- Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Greenplum Database verbinden mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Greenplum Database nach Amazon Redshift zu konvertieren.

Rechte für die Greenplum-Datenbank als Quelle

Für die Verwendung der Greenplum-Datenbank als Quelle sind die folgenden Rechte erforderlich:

- STELLEN SIE EINE VERBINDUNG ZUR DATENBANK HER *<database_name>*
- VERWENDUNG IM SCHEMA *<schema_name>*
- WÄHLEN SIE EIN *<schema_name>.<table_name>*
- WÄHLEN SIE AUF SEQUENZ *<schema_name>.<sequence_name>*

Ersetzen Sie im vorherigen Beispiel die Platzhalter wie folgt:

- *database_name*Ersetzen Sie durch den Namen der Quelldatenbank.
- *schema_name*Ersetzen Sie durch den Namen des Quellschemas.
- *table_name*Ersetzen Sie durch den Namen der Quelltabelle.
- *sequence_name*Ersetzen Sie durch den Namen des Sequenznamens.

Verbindung zur Greenplum-Datenbank als Quelle herstellen

Verwenden Sie das folgende Verfahren, um eine Verbindung zu Ihrer Greenplum-Quelldatenbank herzustellen. AWS SCT

So stellen Sie eine Verbindung zu einer Greenplum-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen.
2. Wählen Sie SAP ASE und dann Weiter.

   Das Dialogfeld Quelle hinzufügen wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Anmeldeinformationen für die Greenplum-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Greenplum-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.

Parameter	Aktion
	AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Vertrauensspeichers zu überprüfen. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

Parameter	Aktion
Treiberpfad für die Greenplum-Datenbank	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Greenplum nach Amazon Redshift

Um die Konvertierungseinstellungen von Greenplum zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Greenplum und dann Greenplum — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Greenplum zu Amazon Redshift an.

Die Konvertierungseinstellungen von Greenplum zu Amazon Redshift AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

  Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

  AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.
- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Einstellungen zur Optimierung der Konvertierung von Greenplum nach Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Greenplum nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Greenplum und dann Greenplum — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Greenplum zu Amazon Redshift an.

Die Einstellungen für die Konvertierungsoptimierung von Greenplum nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

- Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
- Ignoriere Metadaten, verwende statistische Informationen
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
  - Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
  - Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
  - Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

  Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verbindung zu Netezza herstellen mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Netezza nach Amazon Redshift zu konvertieren.

Rechte für Netezza als Quelle

Für die Verwendung von Netezza als Quelle sind die folgenden Rechte erforderlich:

- wählen Sie in der Ansicht system.definition_schema.system
- wählen Sie in der Tabelle system.definition_schema.system
- wählen Sie in der Tabelle system.definition_schema.management
- Liste auf *<database_name>*
- Liste auf *<schema_name>*
- auf *<database_name>* .all.table auflisten
- Liste in der Tabelle .all.external *<database_name>*
- Liste auf .all.view *<database_name>*
- Liste in der Ansicht „.all.materialized“ *<database_name>*
- Liste auf *<database_name>* .all.procedure
- Liste auf .all.sequence *<database_name>*
- Liste auf .all.function *<database_name>*
- Liste auf .all.aggregate *<database_name>*

Ersetzen Sie im vorherigen Beispiel die Platzhalter wie folgt:

- *database_name*Ersetzen Sie durch den Namen der Quelldatenbank.
- *schema_name*Ersetzen Sie durch den Namen des Quellschemas.

AWS SCT erfordert Zugriff auf die folgenden Systemtabellen und Ansichten. Sie können Zugriff auf diese Objekte gewähren, anstatt Zugriff auf `system.definition_schema.system view` und `system.definition_schema.system tables` in der obigen Liste zu gewähren.

- wählen Sie auf system.definition_schema aus. _t_aggregate
- wählen Sie auf system.definition_schema aus. _t_class

- wählen Sie auf system.definition_schema aus. _t_constraint
- wählen Sie auf system.definition_schema aus. _t_const_relattr
- wählen Sie auf system.definition_schema aus. _t_database
- wählen Sie auf system.definition_schema aus. _t_grobj_priv
- wählen Sie auf system.definition_schema aus. _t_grpusr
- wählen Sie auf system.definition_schema aus. _t_hist_config
- wählen Sie auf system.definition_schema aus. _t_object
- wählen Sie auf system.definition_schema aus. _t_object_classes
- wählen Sie auf system.definition_schema aus. _t_proc
- wählen Sie auf system.definition_schema aus. _t_type
- wählen Sie auf system.definition_schema aus. _t_user
- wählen Sie auf system.definition_schema aus. _t_usrobj_priv
- wählen Sie auf system.definition_schema aus. _vt_sequence
- wählen Sie auf system.definition_schema aus. _v_aggregate
- wählen Sie auf system.definition_schema aus. _v_constraint_depends
- wählen Sie auf system.definition_schema aus. _v_database
- wählen Sie auf system.definition_schema aus. _v_datatype
- wählen Sie auf system.definition_schema aus. _v_dslice
- wählen Sie auf system.definition_schema aus. _v_function
- wählen Sie auf system.definition_schema aus. _v_group
- wählen Sie auf system.definition_schema aus. _v_obj_relation
- wählen Sie auf system.definition_schema aus. _v_obj_relation_xdb
- wählen Sie auf system.definition_schema aus. _v_procedure
- wählen Sie auf system.definition_schema aus. _v_relation_column
- wählen Sie auf system.definition_schema aus. _v_relation_keydata
- wählen Sie auf system.definition_schema aus. _v_relobjclasses
- wählen Sie auf system.definition_schema aus. _v_schema_xdb
- wählen Sie auf system.definition_schema aus. _v_sequence
- wählen Sie auf system.definition_schema aus. _v_Synonym
- wählen Sie auf system.definition_schema aus. _v_system_info

- wählen Sie auf system.definition_schema aus. _v_sys_constraint
- wählen Sie auf system.definition_schema aus. _v_sys_object_dslice_info
- wählen Sie auf system.definition_schema aus. _v_sys_user
- wählen Sie auf system.definition_schema aus. _v_table
- wählen Sie auf system.definition_schema aus. _v_table_constraint
- wählen Sie auf system.definition_schema aus. _v_table_dist_map
- wählen Sie auf system.definition_schema aus. _v_table_organize_column
- wählen Sie auf system.definition_schema aus. _v_table_storage_stat
- wählen Sie auf system.definition_schema aus. _v_user
- wählen Sie auf system.definition_schema aus. _v_view
- wählen Sie auf system.information_schema aus. _v_relation_column
- wählen Sie auf system.information_schema aus. _v_table
- wählen Sie auf $hist_column_access_*

Verbindung zu Netezza als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Netezza-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Netezza-Quelldatenbank her

1. Wählen Sie im die AWS Schema Conversion Tool Option Quelle hinzufügen.
2. Wählen Sie Netezza und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Netezza-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

Parameter	Aktion
Netezza-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Konfiguration der fortlaufenden Datenreplikation

Nachdem Sie Ihre Netezza-Datenbankschemas konvertiert und auf Ihre Amazon Redshift Redshift-Datenbank angewendet haben, können Sie Daten mit AWS SCT Datenextraktionsagenten migrieren. Der Agent extrahiert Ihre Daten und lädt sie in Ihren Amazon S3 S3-Bucket hoch. Anschließend können Sie die Daten von Amazon S3 nach Amazon Redshift kopieren. AWS SCT

Wenn sich Daten in Ihrer Quelldatenbank während des Migrationsprozesses ändern, können Sie laufende Änderungen mit Ihren AWS SCT Datenextraktionsagenten erfassen. Anschließend können Sie diese laufenden Änderungen in Ihrer Zieldatenbank replizieren, nachdem Sie die erste Datenmigration abgeschlossen haben. Dieser Vorgang wird als fortlaufende Datenreplikation oder Change Data Capture (CDC) bezeichnet.

So konfigurieren Sie die fortlaufende Datenreplikation für Migrationen von Netezza zu Amazon Redshift

1. Erstellen Sie in Ihrer Quelldatenbank eine Verlaufsdatenbank. Sie können das folgende Codebeispiel in der Netezza-Befehlszeilenschnittstelle (CLI) verwenden.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
 -p your_password
```

Ersetzen Sie es im vorherigen Beispiel *history_database_name* durch den Namen Ihrer Verlaufsdatenbank. Ersetzen Sie es anschließend *load_user* durch den Namen des Benutzers, den Sie für das Laden von Verlaufsdaten in die Datenbank definiert haben. *histdb_owner*Ersetzen Sie es dann durch den Namen des Benutzers, den Sie als Eigentümer der Verlaufsdatenbank definiert haben. Stellen Sie sicher, dass Sie diesen Benutzer bereits erstellt und die `CREATE DATABASE` Erlaubnis erteilt haben. Ersetzen Sie es abschließend *your_password* durch ein sicheres Passwort.

2. Konfigurieren Sie die Verlaufsprotokollierung. Verwenden Sie dazu das folgende Codebeispiel.

```
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;
```

Ersetzen Sie im vorherigen Beispiel *history_configuration_name* und *history_database_name* durch die Namen Ihrer Verlaufskonfiguration und Ihrer Verlaufsdatenbank. Ersetzen Sie es anschließend *load_user* durch den Namen des Benutzers, den Sie für das Laden von Verlaufsdaten in die Datenbank definiert haben. Ersetzen Sie es dann *your_password* durch ein sicheres Passwort.

3. Erteilen Sie Leseberechtigungen für alle Tabellen in der Verlaufsdatenbank. Sie können das folgende Codebeispiel verwenden, um die `SELECT` Berechtigung zu erteilen.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Ersetzen Sie es im vorherigen Beispiel *history_database_name* durch den Namen Ihrer Verlaufsdatenbank. Als Nächstes ersetzen Sie ihn durch *your_user* den Namen des Benutzers mit minimalen Rechten für die Arbeit mit Ihrer Netezza-Datenbank. Sie verwenden die Anmeldeinformationen dieses Datenbankbenutzers in. AWS SCT

4. Sammeln Sie Statistiken für jede Tabelle in Ihrem Quellschema, um Informationen über die Kardinalität der Spalten zu erhalten. Sie können den folgenden Befehl verwenden, um Statistiken in Ihrer Verlaufsdatenbank zu generieren.

```
GENERATE STATISTICS on "schema_name"."table_name";
```

Ersetzen Sie im vorherigen Beispiel *schema_name* und *table_name* durch den Namen Ihres Datenbankschemas und der Tabelle.

5. Stellen Sie sicher, dass Sie die Voraussetzungen erfüllt haben, indem Sie die folgende Abfrage ausführen:

```
SELECT COUNT(*)
 FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Ersetzen Sie im vorherigen Beispiel *history_database_name* und *history_schema_name* durch den Namen Ihrer Verlaufsdatenbank und Ihres Schemas. Als Nächstes *N* ersetzen Sie es durch die Versionsnummer Ihrer Verlaufsdatenbank. Weitere Informationen zu historischen Datenbankversionen finden Sie in der IBM Netezza-Dokumentation.

6. Installieren Sie Ihre Datenextraktionsagenten. Weitere Informationen finden Sie unter Installation von Extraktionsagenten.

Stellen Sie sicher, dass der `{working.folder}` Parameter in der `settings.properties` Datei für alle Extraktor-Instanzen auf denselben Ordner verweist. In diesem Fall können Ihre Extraktoren die CDC-Sitzung koordinieren und einen einzigen Transaktionspunkt für alle Unteraufgaben verwenden.

7. Registrieren Sie Ihren Datenextraktionsagenten. Weitere Informationen finden Sie unter Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool.

8. Erstellen Sie Ihre CDC-Aufgabe. Weitere Informationen finden Sie unter Eine AWS SCT Aufgabe erstellen, ausführen und überwachen.

   a. Öffnen Sie Ihr Projekt in AWS SCT. Wählen Sie im linken Bereich Ihre Quelltabelle aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Lokale Aufgabe erstellen.

   b. Geben Sie unter Aufgabenname einen aussagekräftigen Namen für Ihre Datenmigrationsaufgabe ein.

   c. Wählen Sie für den Migrationsmodus Extrahieren, Hochladen und Kopieren aus.

   d. Wählen Sie CDC aktivieren aus.

   e. Wählen Sie die Registerkarte CDC-Einstellungen und definieren Sie den Umfang und den Zeitplan der CDC-Sitzungen.

   f. Wählen Sie Testaufgabe, um zu überprüfen, ob Sie eine Verbindung zu Ihrem Arbeitsordner, Amazon S3 S3-Bucket und Amazon Redshift Data Warehouse herstellen können.

   g. Wählen Sie Erstellen, um Ihre Aufgabe zu erstellen.

h. Wählen Sie die Registerkarte Aufgaben, wählen Sie Ihre Aufgabe aus der Liste aus und klicken Sie auf Start.

9. Die AWS SCT Aufgabe gewährleistet die Transaktionskonsistenz in der Zieldatenbank. Der Datenextraktionsagent repliziert Transaktionen aus der Quelle in der Reihenfolge der Transaktions-IDs.

   Wenn Sie eine der Migrationssitzungen beenden oder wenn sie fehlschlägt, wird auch die CDC-Verarbeitung gestoppt.

Einstellungen für die Konvertierung von Netezza nach Amazon Redshift

Um die Konvertierungseinstellungen von Netezza zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Netezza und dann Netezza — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Netezza zu Amazon Redshift an.

Die Konvertierungseinstellungen von Netezza zu Amazon Redshift AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen,

die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Einstellungen zur Optimierung der Konvertierung von Netezza nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Netezza zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Netezza und dann Netezza — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Netezza zu Amazon Redshift an.

Die Einstellungen zur Optimierung der Konvertierung von Netezza zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.

- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignoriere Metadaten, verwende statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
  - Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen

Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.

- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Oracle Data Warehouse verbinden mit AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Oracle Data Warehouse nach Amazon Redshift oder Amazon Redshift zu konvertieren und in Kombination zu AWS Glue verwenden.

Rechte für Oracle Data Warehouse als Quelle

Die folgenden Rechte sind erforderlich, um Oracle Data Warehouse als Quelle zu verwenden:

- connect
- select_catalog_role
- select any dictionary

Verbindung zu Oracle Data Warehouse als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit den AWS Schema Conversion Tool eine Verbindung zu Ihrer Oracle Data Warehouse-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Oracle Data Warehouse-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion ToolQuelle hinzufügen aus.
2. Wählen Sie Oracle und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

   - Gehen Sie wie folgt vor, um die Oracle-Quell-Data Warehouse-Verbindungsinformationen manuell einzugeben:

Parameter	Aktion
Typ	Wählen Sie den Verbindungstyp für die Datenbank aus. Geben Sie abhängig vom gewählten Typ die folgenden zusätzlichen Informationen ein: • SID • Servername: Der DNS-Name (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers. • Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird • Oracle SID: Die Oracle-System-ID (SID). Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank: `SELECT sys_context('userenv','instance_name') AS SID FROM dual;` • Service-Name • Server name: Der DNS-Name oder die IP-Adresse Ihres Quelldatenbankservers

Parameter	Aktion
	• Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird • Service Name: Der Name des Oracle-Services, zu dem eine Verbindung hergestellt werden soll • TNS-Alias • TNS file path: Der Pfad zu der Datei, die die Transparent Network Substrate (TNS)-Namensverbindungsinformationen enthält • TNS file path: Der TNS-Alias aus der Datei, die für die Verbindung zur Quelldatenbank verwendet wird • TNS-Connect-ID • TNS Connect-ID: Die Kennung für die registrierten TNS-Verbindungsinformationen.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • SSL-Authentifizierung: Wählen Sie diese Option, um die SSL-Authentifizierung für die Verbindung zu verwenden. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. • Schlüsselspeicher: Der Speicherort eines Schlüssel speichers, der einen privaten Schlüssel und Zertifikate enthält. Dieser Wert ist erforderlich, wenn die SSL-Authe ntifizierung ausgewählt ist, andernfalls ist er optional.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Oracle-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift

Um die Konvertierungseinstellungen von Oracle Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift an.

Die Konvertierungseinstellungen von Oracle Data Warehouse nach Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

  Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

  AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.
- Um Formatierungsfunktionen für Datentypen wie `TO_CHAR``TO_DATE`, und `TO_NUMBER` mit Datetime-Formatelementen zu konvertieren, die Amazon Redshift nicht unterstützt. AWS SCT Verwendet standardmäßig die Funktionen des Erweiterungspakets, um die Verwendung dieser nicht unterstützten Formatelemente im konvertierten Code zu emulieren.

  Das Datetime-Formatmodell in Oracle enthält im Vergleich zu Zeichenketten im Datetime-Format in Amazon Redshift mehr Elemente. Wenn Ihr Quellcode nur Elemente im Datetime-Format enthält, die Amazon Redshift unterstützt, benötigen Sie die Funktionen des Erweiterungspakets im konvertierten Code nicht. Um zu vermeiden, dass die Funktionen des Erweiterungspakets im konvertierten Code verwendet werden, wählen Sie Datetype-Formatelemente aus, die Sie im Oracle-Code verwenden, ähneln Zeichenketten im Datetime-Format in Amazon Redshift. In diesem Fall arbeitet der konvertierte Code schneller.

  Das numerische Formatmodell in Oracle enthält mehr Elemente als numerische Formatzeichenfolgen in Amazon Redshift. Wenn Ihr Quellcode nur numerische Formatelemente enthält, die Amazon Redshift unterstützt, benötigen Sie die Erweiterungspaket-Funktionen im konvertierten Code nicht. Um zu vermeiden, dass die Funktionen des Erweiterungspakets im konvertierten Code verwendet werden, wählen Sie Numerische Formatelemente aus, die Sie im Oracle-Code verwenden, ähneln numerischen Formatzeichenfolgen in Amazon Redshift. In diesem Fall arbeitet der konvertierte Code schneller.

- Um Oracle LEAD - und LAG Analysefunktionen zu konvertieren. Löst standardmäßig ein AWS SCT Aktionselement für jede LEAD LAG AND-Funktion aus.

  Wenn Ihr Quellcode nicht die Standardwerte für Offset in diesen Funktionen verwendet, AWS SCT kann die Verwendung dieser Funktionen mit der NVL Funktion emuliert werden. Wählen Sie dazu die NVL-Funktion verwenden, um das Verhalten der Oracle LEAD- und LAG-Funktionen zu emulieren.
- Um das Verhalten von Primär- und Einzelschlüsseln in Ihrem Amazon Redshift Redshift-Cluster zu emulieren, wählen Sie Das Verhalten von Primär- und Einzelschlüsseln emulieren aus.

  Amazon Redshift erzwingt keine eindeutigen und primären Schlüssel und verwendet sie nur zu Informationszwecken. Wenn Sie diese Einschränkungen in Ihrem Code verwenden, stellen Sie sicher, dass ihr Verhalten im konvertierten Code AWS SCT emuliert wird.
- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Optimierungseinstellungen für die Konvertierung von Oracle Data Warehouse nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Oracle Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Oracle und dann Oracle — Amazon Redshift aus.
Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift an.

Die Einstellungen für die Optimierung der Konvertierung von Oracle Data Warehouse nach Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignorieren Sie Metadaten, verwenden Sie statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.
- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.

- Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verbindung zu einem Snowflake-Data Warehouse herstellen mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Snowflake nach Amazon Redshift zu konvertieren.

Rechte für Snowflake als Quelldatenbank

Sie können eine Rolle mit Rechten erstellen und dieser Rolle den Namen eines Benutzers zuweisen, indem Sie die `SECURITYADMIN` Rolle und den `SECURITYADMIN` Sitzungskontext verwenden.

Das folgende Beispiel erstellt minimale Rechte und gewährt sie dem `min_privs` Benutzer.

```
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;
```

Ersetzen Sie im vorherigen Beispiel die Platzhalter wie folgt:

- *role_name*Ersetzen Sie ihn durch den Namen einer Rolle mit schreibgeschützten Rechten.
- *db_name*Ersetzen Sie durch den Namen der Quelldatenbank.
- *schema_name*Ersetzen Sie durch den Namen des Quellschemas.
- *datawarehousename*Ersetzen Sie durch den Namen eines erforderlichen Data Warehouse.
- *min_privs*Ersetzen Sie es durch den Namen eines Benutzers mit minimalen Rechten.

Die `DEFAULT_WAREHOUSE` Parameter `DEFAULT_ROLE` und sind schlüsselabhängig.

Konfiguration des sicheren Zugriffs auf Amazon S3

Sicherheits- und Zugriffsverwaltungsrichtlinien für einen Amazon S3 S3-Bucket ermöglichen Snowflake den Zugriff auf den S3-Bucket, das Lesen von Daten und das Schreiben von Daten in den S3-Bucket. Sie können den sicheren Zugriff auf einen privaten Amazon S3 S3-Bucket mithilfe des `STORAGE INTEGRATION` Snowflake-Objekttyps konfigurieren. Ein Snowflake-Speicherintegrationsobjekt delegiert die Verantwortung für die Authentifizierung an eine Snowflake-Identitäts- und Zugriffsverwaltungseinheit.

Weitere Informationen finden Sie unter Konfiguration einer Snowflake-Speicherintegration für den Zugriff auf Amazon S3 in der Snowflake-Dokumentation.

Verbindung zu Snowflake als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Quelldatenbank herzustellen. AWS Schema Conversion Tool

So stellen Sie eine Verbindung zu einer Snowflake-Quelldatenbank her

1. Wählen Sie in der die Option AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Snowflake und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen für das Snowflake-Quell-Data Warehouse manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Snowflake-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT speichert Ihr Passwort nur dann in einem verschlüsselten Format, wenn Sie es ausdrücklich anfordern.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Pfad des privaten Schlüssels: Der Speicherort eines privaten Schlüssels. • Passphrase: Die Passphrase für den privaten Schlüssel. Weitere Informationen zur SSL-Unterstützung für Snowflake finden Sie unter Sicherheitsoptionen für Verbindungen konfigurieren.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option festlegen, können Sie das Datenbank kennwort speichern. Auf diese Weise können Sie schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Snowflake-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einschränkungen für Snowflake als Quelle

Die folgenden Einschränkungen gelten bei der Verwendung von Snowflake als Quelle für: AWS SCT

- Objektkennungen müssen im Kontext des Objekttyps und des übergeordneten Objekts eindeutig sein:

  Datenbank

  Schema-IDs müssen innerhalb einer Datenbank eindeutig sein.

  Schemata

  Objektbezeichner, z. B. für Tabellen und Ansichten, müssen innerhalb eines Schemas eindeutig sein.

  Tabellen/Ansichten

  Spaltenbezeichner müssen innerhalb einer Tabelle eindeutig sein.
- Die maximale Anzahl von Tabellen für große und extragroße Cluster-Knotentypen beträgt 9.900. Bei Cluster-Knotentypen vom Typ 8xlarge beträgt die maximale Anzahl von Tabellen 100.000. Das Limit umfasst temporäre Tabellen, sowohl benutzerdefinierte Tabellen als auch Tabellen, die von Amazon Redshift während der Abfrageverarbeitung oder Systemwartung erstellt wurden. Weitere Informationen finden Sie unter Amazon Redshift quotas (Amazon-Redshift-Kontingente) im Amazon-Redshift-Clusterverwaltungshandbuch.
- Für gespeicherte Prozeduren beträgt die maximale Anzahl von Eingabe- und Ausgabeargumenten 32.

Quelldatentypen für Snowflake

Im Folgenden finden Sie die Snowflake-Quelldatentypen, die bei der Verwendung unterstützt werden, AWS SCT und die Standardzuordnung zu einem Amazon Redshift Redshift-Ziel.

Snowflake-Datentypen	Amazon-Redshift-Datentypen
NUMBER	NUMERISCH (38)
NUMBER(p)	Wenn p =< 4 ist, dann SMALLINT Wenn p => 5 und =< 9 ist, dann INTEGER

Snowflake-Datentypen	Amazon-Redshift-Datentypen
	Wenn p => 10 und =< 18 ist, dann BIGINT Wenn p => 19 ist, dann NUMERIC (p)
ZAHL (p, 0)	Wenn p =< 4 ist, dann SMALLINT Wenn p => 5 und =< 9 ist, dann INTEGER Wenn p => 10 und =< 18 ist, dann BIGINT Wenn p => 19 ist, dann: NUMERIC (p,0)
ZAHL (p, s)	Wenn p => 1 und =< 38 ist und wenn s => 1 und =< 37 ist, dann NUMERIC (p,s)
FLOAT	FLOAT
TEXT Unicode-Zeichen bis zu 16.777.216 Byte; bis zu 4 Byte pro Zeichen.	VARCHAR(MAX)
TEXT (p) Unicode-Zeichen bis zu 65.535 Byte; bis zu 4 Byte pro Zeichen.	Wenn p =< 65.535 ist, dann ist VARCHAR (p)
TEXT (p) Unicode-Zeichen bis zu 16.777.216 Byte; bis zu 4 Byte pro Zeichen.	Wenn p => 65.535 und =< 16.777.216 ist, dann ist VARCHAR (MAX)
BINARY Einzelbyte-Zeichen bis zu 8.388.608 Byte; 1 Byte pro Zeichen.	VARCHAR(MAX)

Snowflake-Datentypen	Amazon-Redshift-Datentypen
BINÄR (p) Einzelbyte-Zeichen bis zu 65.535 Byte; 1 Byte pro Zeichen.	VARCHAR (p)
BINÄR (p) Einzelbyte-Zeichen bis zu 8.388.608 Byte; 1 Byte pro Zeichen.	VARCHAR(MAX)
BOOLEAN	BOOLEAN
DATUM	DATUM
TIME Zeitwerte zwischen 00:00:00 und 23:59:59.999 999999.	VARCHAR(18)
ZEIT (f) Zeitwerte zwischen 00:00:00 und 23:59:59.9 (f).	VARCHAR (n) — 9 + dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP (ZEITSTEMPEL)
ZEITSTEMPEL_TZ	TIMESTAMPTZ

Einstellungen für die Konvertierung von Snowflake nach Amazon Redshift

Um die Konvertierungseinstellungen von Snowflake nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Snowflake und dann Snowflake — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Snowflake zu Amazon Redshift an.

Die Konvertierungseinstellungen von Snowflake zu Amazon Redshift AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung

auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Einstellungen für die Optimierung der Konvertierung von Snowflake zu Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Snowflake zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Snowflake und dann Snowflake — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Snowflake zu Amazon Redshift an.

Die Einstellungen zur Optimierung der Konvertierung von Snowflake zu Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:
  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignorieren Sie Metadaten, verwenden Sie statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl

der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
  - Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
  - Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
  - Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

  Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Herstellen einer Verbindung zu einem SQL Server Data Warehouse mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Microsoft SQL Server DW nach Amazon Redshift oder Amazon Redshift zu konvertieren und in Kombination zu AWS Glue verwenden.

Rechte für Microsoft SQL Server Data Warehouse als Quelle

Die folgenden Rechte sind erforderlich, um Microsoft SQL Server Data Warehouse als Quelle zu verwenden:

- VIEW DEFINITION
- VIEW DATABASE STATE
- WÄHLEN SIE EIN SCHEMA AUS: *<schema_name>*

Ersetzen Sie im vorherigen Beispiel den *<source_schema>* Platzhalter durch den Namen der Quelle source_schema.

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Gewähren Sie darüber hinaus folgende Berechtigungen für die Master-Datenbank:

- VIEW SERVER STATE

Einschränkungen für SQL Server Data Warehouse als Quelle

Die Verwendung von Microsoft SQL Server Parallel Data Warehouse (PDW) als Quelle wird derzeit nicht unterstützt.

Eine Verbindung zu SQL Server Data Warehouse als Quelle herstellen

Verwenden Sie das folgende Verfahren, um eine Verbindung zu Ihrer SQL Server Data Warehouse-Quelldatenbank mit dem herzustellen AWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer SQL Server Data Warehouse-Quelldatenbank her

1. Wählen Sie in der die AWS Schema Conversion Tool Option Quelle hinzufügen aus.
2. Wählen Sie Microsoft SQL Server und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:

  1. Wählen Sie für AWS Secret den Namen des Secrets aus.
  2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

  Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

- Gehen Sie wie folgt vor, um die Microsoft SQL Server-Quell-Data Warehouse-Verbindungsinformationen manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Instance name	Geben Sie den Instanznamen für das SQL Server-Data Warehouse ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat vertrauen: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen. • Vertrauensspeicher: Ein Vertrauensspeicher, den Sie in den globalen Einstellungen eingerichtet haben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Pfad des SQL Server-Tr eibers	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von SQL Server Data Warehouse nach Amazon Redshift

Um die Konvertierungseinstellungen von SQL Server Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus.

Wählen Sie in der oberen Liste Microsoft SQL Server und dann Microsoft SQL Server — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server Data Warehouse zu Amazon Redshift an.

Die Konvertierungseinstellungen von SQL Server Data Warehouse zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.
- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Optimierungseinstellungen für die Konvertierung von SQL Server Data Warehouse nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von SQL Server Data Warehouse in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Microsoft SQL Server und dann Microsoft SQL Server — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von SQL Server Data Warehouse zu Amazon Redshift an.

Die Einstellungen für die Optimierung der Konvertierung von SQL Server Data Warehouse nach Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignorieren Sie Metadaten, verwenden Sie statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.
- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.

- Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Herstellen einer Verbindung zu einem Teradata Data Warehouse mit dem AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Teradata nach Amazon Redshift oder Amazon Redshift zu konvertieren und in Kombination zu verwenden. AWS Glue

Rechte für Teradata als Quelle

Für die Verwendung von Teradata als Quelle sind die folgenden Rechte erforderlich:

- SELECT ON DBC
- WÄHLEN SIE AUF SYSUDTLIB
- WÄHLEN SIE AUF SYSLIB
- WÄHLEN SIE EIN *<source_database>*
- PROZEDUR ERSTELLEN AM *<source_database>*

Ersetzen Sie im vorherigen Beispiel den *<source_database>* Platzhalter durch den Namen der Quelldatenbank.

AWS SCT erfordert das CREATE PROCEDURE-Privileg, um HELP PROCEDURE für alle Prozeduren in der Quelldatenbank auszuführen. AWS SCT verwendet dieses Recht nicht, um neue Objekte in Ihrer Teradata-Quelldatenbank zu erstellen.

Verbindung zu Teradata als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Teradata-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Teradata-Quelldatenbank her

1. Wählen Sie im die Option AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Teradata und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

     Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.
   - Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Teradata-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Verbindungsname	Geben Sie einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.

Parameter	Aktion
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Teradata-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Daten verschlüsseln	Wählen Sie diese Option, um Daten zu verschlüsseln, die Sie mit der Datenbank austauschen. Wenn Sie diese Option wählen, wird die Portnummer 443 für die Übertragu ng verschlüsselter Daten zwischen AWS SCT und Ihrer Teradata-Datenbank verwendet.

Parameter	Aktion
Teradata-Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Verwenden der LDAP-Authentifizierung mit einer Teradata-Quelle

Gehen Sie wie folgt vor, um eine Lightweight Directory Access Protocol (LDAP)-Authentifizierung für Teradata-Benutzer einzurichten, die Microsoft Active Directory in Windows verwenden.

Im folgenden Verfahren ist die Active Directory-Domäne. `test.local.com` Der Windows-Server ist esDC, und er ist mit Standardeinstellungen konfiguriert. Das folgende Skript erstellt das `test_ldap` Active Directory-Konto, und dieses Konto verwendet das `test_ldap` Passwort.

So richten Sie die LDAP-Authentifizierung für Teradata-Benutzer ein, die Microsoft Active Directory in Windows ausführen

1. Bearbeiten Sie im Verzeichnis `/opt/teradata/tdat/tdgss/site` die Datei `TdgssUserConfigFile.xml`. Ändern Sie den LDAP-Abschnitt wie folgt.

```
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. Übernehmen Sie die Änderungen, indem Sie die Konfiguration wie folgt ausführen.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Testen Sie die Konfiguration, indem Sie den folgenden Befehl ausführen.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen.

```
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. Starten Sie TPA mit dem folgenden Befehl neu.

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. Erstellen Sie, wie im Folgenden gezeigt, denselben Benutzer in der Teradata-Datenbank wie in Active Directory.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Wenn Sie das Benutzerkennwort in Active Directory für Ihren LDAP-Benutzer ändern, geben Sie dieses neue Passwort bei der Verbindung zu Teradata im LDAP-Modus an. Im DEFAULT-Modus

stellen Sie mithilfe des LDAP-Benutzernamens und eines beliebigen Kennworts eine Verbindung zu Teradata her.

Konfiguration der Statistikerfassung in Ihrem Teradata-Quell-Data Warehouse

Um Ihr Teradata-Quell-Data Warehouse zu konvertieren, AWS SCT verwendet Statistiken, um Ihr konvertiertes Amazon Redshift Redshift-Data Warehouse zu optimieren. Sie können Statistiken in der Statistikdatei sammeln AWS SCT oder diese hochladen. Weitere Informationen finden Sie unter Statistiken sammeln oder hochladen.

Um sicherzustellen, dass Statistiken aus Ihrem Data Warehouse erfasst werden AWS SCT können, müssen Sie die folgenden erforderlichen Aufgaben ausführen.

Um Statistiken aus Ihrem Teradata Data Warehouse zu sammeln

1. Führen Sie die folgende Abfrage aus, um Statistiken für alle Tabellen in Ihrem Data Warehouse abzurufen.

   ```
   collect summary statistics on table_name;
   ```

   Ersetzen Sie im vorherigen Beispiel durch *table_name* den Namen Ihrer Quelltabelle. Wiederholen Sie die Abfrage für jede Tabelle, die Sie konvertieren.
2. Führen Sie die folgende Abfrage aus, um die Kontozeichenfolge für den Benutzer zu ermitteln, mit der Sie Ihr Data Warehouse konvertieren.

   ```
   select * from dbc.accountinfo where username ='user_name'
   ```
3. Aktivieren Sie die Abfrageprotokollierung für einen bestimmten Benutzer mithilfe der Kontozeichenfolge aus dem vorherigen Beispiel.

   ```
   BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('$M$BUSI$S$D$H');
   ```

   Alternativ können Sie die Abfrageprotokollierung für alle Datenbankbenutzer aktivieren.

   ```
   BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
   ```

Nachdem Sie die Erfassung der Data Warehouse-Statistiken abgeschlossen haben, deaktivieren Sie die Abfrageprotokollierung. Dazu können Sie das folgende Codebeispiel verwenden.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$S$D$H');
```

Sammeln von Statistiken im Offline-Modus aus Ihrem Teradata-Quell-Data Warehouse

Nachdem Sie die Statistikerfassung in Ihrem Teradata Data Warehouse konfiguriert haben, können Sie Statistiken in Ihrem AWS SCT Projekt sammeln. Alternativ können Sie Basic Teradata Query-Skripts (BTEQ) verwenden, um Statistiken im Offline-Modus zu sammeln. Anschließend können Sie die Dateien mit den gesammelten Statistiken in Ihr Projekt hochladen. AWS SCT Weitere Informationen finden Sie unter Statistiken sammeln oder hochladen.

Um Statistiken aus Ihrem Teradata Data Warehouse im Offline-Modus zu sammeln

1. Erstellen Sie das `off-line_stats.bteq` Skript mit dem folgenden Inhalt.

```
.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_coun
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','BO')
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.dbqlsqltbl 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. Erstellen Sie die `td_run_bteq.bat` Datei, die das BTEQ-Skript ausführt, das Sie im vorherigen Schritt erstellt haben. Verwenden Sie den folgenden Inhalt für diese Datei.

```
@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. Erstellen Sie die `runme.bat` Datei, in der die Batch-Datei ausgeführt wird, die Sie im vorherigen Schritt erstellt haben. Verwenden Sie den folgenden Inhalt für diese Datei.

   ```
   .\td_run_bteq.bat ServerName UserName DatabaseName
   ```

   Ersetzen Sie in der `runme.bat` Datei *`ServerNameUserName`*, und *`DatabaseName`* durch Ihre zutreffenden Werte.

   Führen Sie dann die `runme.bat` Datei aus. Wiederholen Sie diesen Schritt für jedes Data Warehouse, das Sie in Amazon Redshift konvertieren.

Nachdem Sie dieses Skript ausgeführt haben, erhalten Sie drei Dateien mit Statistiken für jede Datenbank. Sie können diese Dateien in Ihr AWS SCT Projekt hochladen. Wählen Sie dazu im linken Bereich Ihres Projekts Ihr Data Warehouse aus und öffnen Sie das Kontextmenü (Rechtsklick). Wählen Sie Statistik hochladen.

Einstellungen für die Konvertierung von Teradata nach Amazon Redshift

Um die Konvertierungseinstellungen von Teradata nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Teradata und dann Teradata — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Teradata zu Amazon Redshift an.

Die Konvertierungseinstellungen von Teradata nach Amazon Redshift AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.
- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

  Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.
- Um eine explizite Liste von Spalten im konvertierten Code für `SELECT *` Anweisungen zu verwenden, wählen Sie Explizite Spaltendeklaration verwenden aus.
- Um das Verhalten von Primär- und Einzelschlüsseln in Ihrem Amazon Redshift Redshift-Cluster zu emulieren, wählen Sie Das Verhalten von Primär- und Einzelschlüsseln emulieren aus.

  Amazon Redshift erzwingt keine eindeutigen und primären Schlüssel und verwendet sie nur zu Informationszwecken. Wenn Sie diese Einschränkungen in Ihrem Code verwenden, stellen Sie sicher, dass ihr Verhalten im konvertierten Code AWS SCT emuliert wird.
- Um die Einzigartigkeit der Daten in den Amazon Redshift Redshift-Zieltabellen sicherzustellen. Wählen Sie dazu das Verhalten von SET-Tabellen emulieren aus.

  Teradata erstellt Tabellen mit dem `SET` Syntaxelement als Standardoption. Sie können einer `SET` Tabelle keine doppelten Zeilen hinzufügen. Wenn Ihr Quellcode diese Eindeutigkeitsbeschränkung nicht verwendet, schalten Sie diese Option aus. In diesem Fall arbeitet der konvertierte Code schneller.

  Wenn Ihr Quellcode die `SET` Option in Tabellen als Eindeutigkeitsbeschränkung verwendet, aktivieren Sie diese Option. In diesem Fall werden die `INSERT..SELECT` Anweisungen im

konvertierten Code AWS SCT neu geschrieben, um das Verhalten Ihrer Quelldatenbank zu emulieren.

Einstellungen zur Optimierung der Konvertierung von Teradata nach Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Teradata in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Teradata und dann Teradata — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Teradata zu Amazon Redshift an.

Die Einstellungen für die Optimierung der Konvertierung von Teradata zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:
  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignorieren Sie Metadaten, verwenden Sie statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

  - Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
  - Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
  - Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
  - Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

  Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verbindung der mit den AWS Schema Conversion Tool Vertica-Datenbanken herstellen

Sie können AWS SCT damit Schemas, Codeobjekte und Anwendungscode von Vertica nach Amazon Redshift konvertieren.

Rechte für Vertica als Quelle

Für die Verwendung von Vertica als Quelle sind die folgenden Rechte erforderlich:

- VERWENDUNG AUF SCHEMA *<schema_name>*
- USAGE ON SCHEMA PUBLIC
- WÄHLEN SIE FÜR ALLE TABELLEN IM SCHEMA *<schema_name>*
- WÄHLEN SIE ALLE SEQUENZEN IM SCHEMA AUS *<schema_name>*
- FÜR ALLE FUNKTIONEN IM SCHEMA AUSFÜHREN *<schema_name>*
- BEI PROZEDUR AUSFÜHREN *<schema_name.procedure_name(procedure_signature)>*

Ersetzen Sie im vorherigen Beispiel die Platzhalter wie folgt:

- *schema_name*Ersetzen Sie durch den Namen des Quellschemas.
- Durch *procedure_name* den Namen einer Quellprozedur ersetzen. Wiederholen Sie den Zuschuss für jedes Verfahren, das Sie konvertieren.
- Ersetzen Sie es *procedure_signature* durch die kommagetrennte Liste der Argumenttypen von Prozeduren.

Verbindung zu Vertica als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Vertica-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Vertica-Quelldatenbank her

1. Wählen Sie im AWS Schema Conversion Tool die Option Quelle hinzufügen.
2. Wählen Sie Vertica und dann Weiter.

   Das Dialogfeld „Quelle hinzufügen“ wird angezeigt.
3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankanmeldedaten von AWS Secrets Manager oder geben Sie sie manuell ein:

   - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
     1. Wählen Sie für AWS Secret den Namen des Secrets aus.
     2. Wählen Sie Auffüllen, um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldedaten von Secrets Manager finden Sie unterKonfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Vertica-Quelldatenbank manuell einzugeben:

Parameter	Aktion
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Vertica-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldedaten ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert das Passwort standardmäßig AWS SCT nicht. Wenn Sie Ihr AWS SCT -Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.

Parameter	Aktion
Use SSL	Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie auf der Registerkarte SSL gegebenenfalls die folgenden zusätzlichen Informationen ein: • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Trust Stores zu überprüfen. • Trust Store: Ein Trust Store, den Sie in den globalen Einstellungen eingerichtet haben. • Schlüsselspeicher: Ein Schlüsselspeicher, den Sie in den globalen Einstellungen eingerichtet haben.
Passwort speichern	AWS SCT erstellt einen sicheren Tresor zum Speichern von SSL-Zertifikaten und Datenbankkennwörtern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Vertikaler Treiberpfad	Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool. Wenn Sie den Treiberpfad in den globalen Projektei nstellungen speichern, wird der Pfad im Verbindungsdialogf eld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Vertica nach Amazon Redshift

Um die Konvertierungseinstellungen von Vertica in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Vertica und dann Vertica — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Vertica zu Amazon Redshift an.

Die Konvertierungseinstellungen von Vertica zu Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

  Wählen Sie für Hinzufügen von Kommentaren zum konvertierten Code für Aktionselemente mit ausgewähltem Schweregrad und höherem Schweregrad den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

  Beispiel: Um die Anzahl der Kommentare im konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in den konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten aus.
- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift Redshift-Zielcluster gelten AWS SCT können.

  Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift Redshift-Zielcluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift Redshift-Cluster gelten AWS SCT können.

  Amazon Redshift verfügt über Kontingente, die die Verwendungstabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt die Anzahl der Tabellen, die auf Ihren Amazon Redshift Redshift-Zielcluster angewendet werden sollen, je nach Knotentyp. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter Kontingente und Limits in Amazon Redshift im Verwaltungshandbuch zu Amazon Redshift.

  AWS SCT konvertiert all Ihre Quelltabellen, auch wenn das mehr ist, als Ihr Amazon Redshift Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie beim Anwenden des konvertierten Codes das Amazon Redshift Redshift-Clusterkontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT Wendet außerdem Tabellen auf Ihren Amazon Redshift Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltabelle in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltabelle erstellt werden AWS SCT können.

  Amazon Redshift unterstützt nicht die Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltabelle in eine separate Tabelle in Amazon Redshift migrieren. AWS SCT Erstellt dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

  AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltabelle. Je nach Art der Quelltabellenpartitionierung kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon-Redshift-Cluster anwenden können. Um zu verhindern, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltabelle erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage im Jahr und zwei Tabellen für `NO RANGE` und `UNKNOWN` -Partitionen entspricht.
- Um die Komprimierung auf Amazon Redshift Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu Kompressionskodierung verwenden aus.

  AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift Redshift-Algorithmus automatisch eine Komprimierungskodierung zu. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.

  Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Komprimierungskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Komprimierungskodierung verwenden auswählen.

Einstellungen für die Optimierung der Konvertierung von Vertica nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Vertica zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen aus. Wählen Sie in der oberen Liste Vertica und dann Vertica — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Vertica zu Amazon Redshift an.

Die Einstellungen zur Optimierung der Konvertierung von Vertica zu Amazon Redshift AWS SCT umfassen Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu Automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

  Die automatische Tabellenoptimierung ist ein sich selbst optimierender Prozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter Arbeiten mit automatischer Tabellenoptimierung im Amazon Redshift Database Developer Guide.

  Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl.
- Um Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

  Sie können Sortier- und Verteilungsschlüssel mithilfe von Amazon Redshift Redshift-Metadaten, statistischen Informationen oder beiden Optionen auswählen. Wählen Sie für Strategie zur ersten Schlüsselauswahl auf der Registerkarte Optimierungsstrategien eine der folgenden Optionen:

  - Verwenden Sie Metadaten, ignorieren Sie statistische Informationen
  - Ignorieren Sie Metadaten, verwenden Sie statistische Informationen
  - Verwenden Sie Metadaten und statistische Informationen

  Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. Anhand dieser Gewichtungswerte AWS SCT wird definiert, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

  Sie können die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.
- Um Strategiedetails zu konfigurieren.

  Sie können nicht nur die Gewichtung für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung aus.

- Geben Sie unter Limit für Sortierschlüsselspalten die maximale Anzahl von Spalten im Sortierschlüssel ein.
- Geben Sie für Schrägschwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, bei denen der Schrägwert größer als der Schwellenwert ist, aus der Kandidatenliste für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des am häufigsten vorkommenden Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Außerdem können Sie auf der Registerkarte Optimierungsstrategien die Größe kleiner Tabellen für die Strategie „Kleine Tabellen suchen“ definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die Mindest- und Höchstanzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den `ALL` Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Zuordnung von Datentypen in der AWS Schema Conversion Tool

Sie können mehrere Quell- und Zieldatenbanken in einem einzigen AWS SCT Projekt hinzufügen. Dies vereinfacht die Verwaltung von Projekten, wenn Sie mehrere Datenbanken auf verschiedene Zielplattformen migrieren.

Nachdem Sie ein neues Projekt erstellt und Quell- und Zieldatenbanken hinzugefügt haben, erstellen Sie Zuordnungsregeln. AWS SCT erfordert mindestens eine Zuordnungsregel, um einen Migrationsbewertungsbericht zu erstellen und Datenbankschemas zu konvertieren.

Eine Zuordnungsregel beschreibt ein Quell-Ziel-Paar, das ein Quelldatenbankschema oder eine Quelldatenbank und eine Zieldatenbankplattform umfasst. Sie können mehrere Zuordnungsregeln in einem einzigen AWS SCT Projekt erstellen. Verwenden Sie Zuordnungsregeln, um jedes Quelldatenbankschema in die richtige Zieldatenbankplattform zu konvertieren.

Um den Namen Ihres Schemas im konvertierten Code zu ändern, richten Sie eine Migrationsregel ein. Mit Migrationsregeln können Sie beispielsweise Ihr Schema umbenennen, Objektnamen ein Präfix hinzufügen, die Spaltensortierung ändern oder Datentypen ändern. Um diese Änderungen auf Ihren konvertierten Code anzuwenden, stellen Sie sicher, dass Sie Migrationsregeln erstellen, bevor Sie Ihr Quellschema konvertieren. Weitere Informationen finden Sie unter Anwendung von Migrationsregeln.

Sie können Zuordnungsregeln nur für unterstützte Datenbankkonvertierungspaare erstellen. Eine Liste der unterstützten Konvertierungspaare finden Sie unterHerstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool.

Wenn Sie ein in AWS SCT Version 1.0.655 oder früher gespeichertes Projekt öffnen, AWS SCT werden automatisch Zuordnungsregeln für alle Quelldatenbankschemas zur Zieldatenbankplattform erstellt. Um weitere Zieldatenbankplattformen hinzuzufügen, löschen Sie bestehende Zuordnungsregeln und erstellen Sie dann neue Zuordnungsregeln.

Themen

- Zuordnung neuer Datentypen in AWS Schema Conversion Tool
- Bearbeiten von Datentypzuordnungen in AWS Schema Conversion Tool
- Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool

- Einschränkungen der Datentypzuordnung in AWS Schema Conversion Tool

Zuordnung neuer Datentypen in AWS Schema Conversion Tool

Sie können mehrere Zuordnungsregeln in einem einzigen Projekt erstellen. AWS SCT speichert Zuordnungsregeln als Teil Ihres Projekts. Gehen Sie bei geöffnetem Projekt wie folgt vor, um eine neue Zuordnungsregel hinzuzufügen.

So erstellen Sie Zuweisungsregeln

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie im linken Bereich ein Schema oder eine Datenbank aus, die der Zuordnungsregel hinzugefügt werden soll.
3. Wählen Sie im rechten Bereich eine Zieldatenbankplattform für das ausgewählte Quellschema oder die ausgewählte Quelldatenbank aus.

   Sie können eine virtuelle Datenbankplattform als Ziel wählen. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
4. Wählen Sie Zuordnung erstellen.

   AWS SCT fügt diese neue Zuordnungsregel zur Liste der Serverzuordnungen hinzu.

Fügt Zuordnungsregeln für alle Konvertierungspaare hinzu. Um einen Bewertungsbericht zu erstellen oder Datenbankschemas zu konvertieren, wählen Sie im Menü Ansicht die Option Hauptansicht.

AWS SCT hebt alle Schemaobjekte, die Teil einer Zuordnungsregel sind, fett hervor.

Bearbeiten von Datentypzuordnungen in AWS Schema Conversion Tool

Sie können vorhandene Zuordnungsregeln filtern oder löschen und Ihrem AWS Schema Conversion Tool () -Projekt eine neue Zuordnungsregel hinzufügen.AWS SCT

Wenn Sie eine Zuordnungsregel für die gesamte Quelldatenbank erstellen, AWS SCT wird für jedes Quelldatenbankschema eine Zuordnungsregel erstellt. Bei Projekten, die Dutzende von Schemas oder sogar Datenbanken beinhalten, kann es schwierig sein zu verstehen, welches Ziel für ein

bestimmtes Schema verwendet wird. Um schnell eine Zuordnungsregel für Ihr Schema zu finden, verwenden Sie eine oder mehrere der folgenden Filteroptionen in AWS SCT.

Um Zuordnungsregeln zu filtern

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie für Quellserver die Quelldatenbank aus.

   Der Filter ist standardmäßig auf Alle eingestellt, was bedeutet, dass Zuordnungsregeln für alle Quelldatenbanken AWS SCT angezeigt werden.
3. Geben Sie für Quellschema den Namen des Quellschemas ein. Verwenden Sie den Prozentsatz (%) als Platzhalter, um eine beliebige Anzahl von Symbolen im Schemanamen zu ersetzen.

   Der Standardfilter ist der Platzhalter%, was bedeutet, dass Zuordnungsregeln für alle Schemanamen der Quelldatenbank AWS SCT angezeigt werden.
4. Wählen Sie für Hat Migrationsregeln die Option Ja aus, um die Zuordnungsregeln anzuzeigen, für die die Datenmigrationsregeln erstellt wurden. Wählen Sie Nein, um Zuordnungsregeln anzuzeigen, für die es keine Datenmigrationsregeln gibt. Weitere Informationen finden Sie unter Regeln für die Datenmigration erstellen in AWS SCT.

   Der Filter ist standardmäßig auf Alle eingestellt, was bedeutet, dass alle Zuordnungsregeln AWS SCT angezeigt werden.
5. Wählen Sie für Zielserver die Zieldatenbank aus.

   Der Filter ist standardmäßig auf Alle eingestellt, was bedeutet, dass Zuordnungsregeln für alle Zieldatenbanken AWS SCT angezeigt werden.

Gehen Sie bei geöffnetem Projekt wie folgt vor, um eine Zuordnungsregel zu löschen. Weitere Informationen zum Hinzufügen von Zuordnungsregeln finden Sie unterZuordnung neuer Datentypen in AWS Schema Conversion Tool.

So löschen Sie Zuordnungsregeln

1. Wählen Sie im Menü Ansicht die Option Zuordnungsansicht aus.
2. Wählen Sie für Serverzuordnungen die Zuordnungsregeln aus, die gelöscht werden sollen.
3. Wählen Sie Ausgewählte Zuordnungen löschen aus.

   AWS SCT löscht die ausgewählten Zuordnungsregeln.

Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool

Sie können sehen, wie Ihr Quelldatenbankschema in jede unterstützte Zieldatenbankplattform AWS SCT konvertiert wird. Dazu müssen Sie keine Verbindung zu einer vorhandenen Zieldatenbank herstellen. Stattdessen können Sie beim Erstellen einer Zuordnungsregel im rechten Bereich eine virtuelle Zieldatenbankplattform auswählen. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool. Stellen Sie sicher, dass Sie die Server, NoSQL-Cluster und ETL-Knoten im rechten Bereich erweitern, um die Liste der virtuellen Zieldatenbankplattformen zu sehen.

AWS SCT unterstützt die folgenden virtuellen Zieldatenbankplattformen:

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon-DynamoDB
- Amazon Redshift
- Amazon Redshift und AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish für Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Wenn Sie Babelfish for Aurora PostgreSQL als Zieldatenbankplattform verwenden, können Sie nur einen Bewertungsbericht zur Datenbankmigration erstellen. Weitere Informationen finden Sie unter the section called “Bewertungsbericht”.

Wenn Sie eine virtuelle Zieldatenbankplattform verwenden, können Sie konvertierten Code in einer Datei speichern. Weitere Informationen finden Sie unter the section called “Speichern des konvertierten Schemas”.

Einschränkungen der Datentypzuordnung in AWS Schema Conversion Tool

Bei der Konvertierung von Schemas mit mehreren Servern in einem einzigen Projekt gelten die folgenden Einschränkungen: AWS SCT

- Sie können denselben Server nur einmal zu einem Projekt hinzufügen.
- Sie können Serverschemas nicht einem bestimmten Zielschema zuordnen, sondern nur einem Zielserver. AWS SCT erstellt das Zielschema während der Konvertierung.
- Sie können dem Zielserver keine Quellobjekte auf niedrigerer Ebene zuordnen.
- Sie können ein Quellschema nur einem Zielserver in einem Projekt zuordnen.
- Stellen Sie sicher, dass Sie einem Zielserver eine Quelle zuordnen, um einen Bewertungsbericht zu erstellen, Schemas zu konvertieren oder Daten zu extrahieren.

Arbeiten mit Berichten in der AWS Schema Conversion Tool

Wenn Sie die Konvertierung einer Datenbank planen, ist es hilfreich, einige Berichte erstellen. So können Sie besser überblicken, welche Aktivitäten erforderlich sind. Sie können Berichte erstellen mit. AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um einen Bewertungsbericht zur Datenbankmigration zu erstellen. Dieser Bericht fasst alle Aufgaben der Schemakonvertierung zusammen und führt detailliert die Elemente auf, die nicht automatisch in die Zieldatenbank konvertiert werden können. Anhand dieses Berichts können Sie auswerten, welcher Teil des Projekts mit der Verwendung AWS SCT abgeschlossen werden kann und was Sie sonst noch für die Konvertierung benötigen. Verwenden Sie zum Erstellen eines Bewertungsberichts die Option Create Report (Bericht erstellen) aus dem Kontextmenü (rechte Maustaste) der Datenbank in AWS SCT.

Themen

- Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool

Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool

Ein wichtiger Teil von AWS Schema Conversion Tool ist der Bewertungsbericht, der generiert wird, um die Komplexität Ihrer Schemakonvertierung einzuschätzen. Dieser Bewertungsbericht zur Datenbankmigration fasst alle Aufgaben zur Schemakonvertierung zusammen und beschreibt detailliert die Aktionspunkte für Schemas, die nicht in die DB-Engine Ihrer Ziel-DB-Instance konvertiert werden können. Sie können den Bericht in der Anwendung anzeigen oder ihn als kommagetrennte Datei (CSV) oder als PDF-Datei exportieren.

Wenn Sie mehrere Quell- und Zieldatenbanken in einem einzigen Projekt hinzufügen, AWS SCT fasst die Berichte für alle Konvertierungspaare in einem Bewertungsbericht zur Datenbankmigration zusammen.

Sie können virtuelle Zieldatenbankplattformen verwenden, um einen Bewertungsbericht zu erstellen und sich ein Bild von der Komplexität der Migration zu einer ausgewählten Datenbankplattform zu machen. In diesem Fall müssen Sie keine Verbindung zu Ihrer Zieldatenbankplattform herstellen. Sie können Babelfish for Aurora PostgreSQL beispielsweise als virtuelle Zieldatenbankplattform

verwenden, um einen Bewertungsbericht zur Datenbankmigration zu erstellen. Weitere Informationen zu virtuellen Zieldatenbankplattformen finden Sie unter. the section called “Zuordnung virtueller Ziele”

Der Bericht zur Bewertung der Datenbankmigration enthält die folgenden Elemente:

- Zusammenfassung
- Lizenzauswertung
- Cloud-Support, mit Angabe aller Funktionen in der Quelldatenbank, die auf dem Ziel nicht verfügbar sind.
- Empfehlungen, einschließlich Konvertierung von Server-Objekten, Backup-Vorschläge und verknüpfte Server-Änderungen

Außerdem enthält der Bericht eine Aufwandsschätzung, da für die nicht automatisch konvertierbaren Elemente entsprechender Code für die DB-Ziel-Instance geschrieben werden muss.

Wenn Sie Ihr vorhandenes Schema AWS SCT auf eine Amazon RDS-DB-Instance migrieren, können Sie den Bericht verwenden, um die Anforderungen für die Migration in die AWS Cloud zu analysieren und Ihren Lizenztyp zu ändern.

Themen

- Erstellen eines Bewertungsberichts in AWS Schema Conversion Tool
- Einen Bewertungsbericht anzeigen in AWS Schema Conversion Tool
- Speichern Sie den Bewertungsbericht in AWS Schema Conversion Tool
- Konfiguration eines Bewertungsberichts in AWS Schema Conversion Tool
- Einen Bewertungsbericht für mehrere Server erstellen in AWS Schema Conversion Tool

Erstellen eines Bewertungsberichts in AWS Schema Conversion Tool

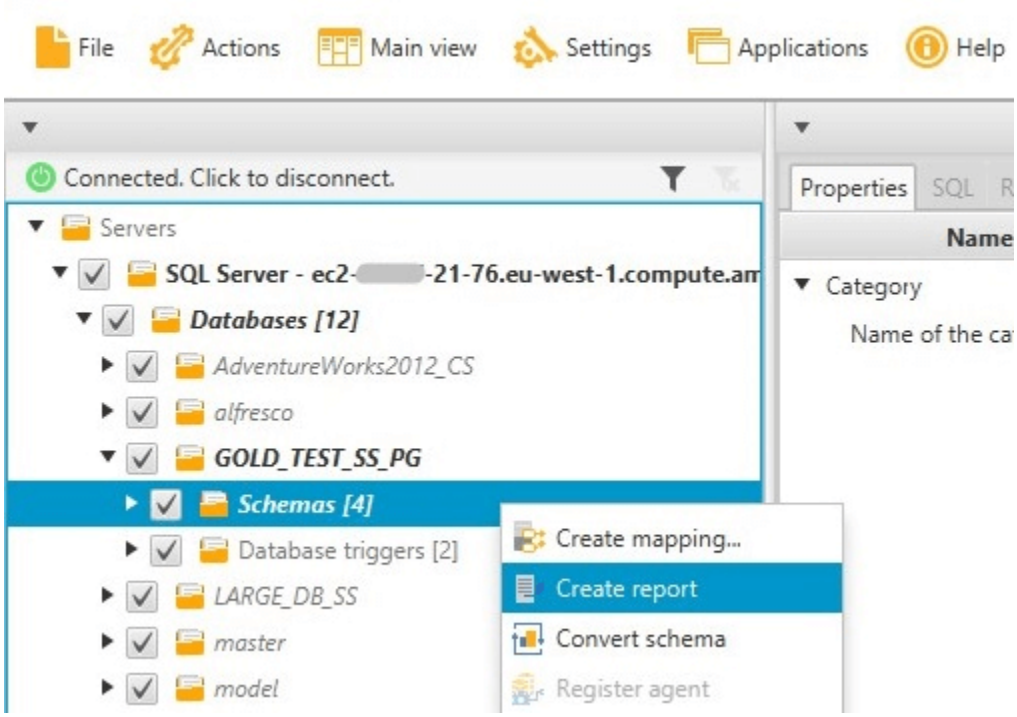
Führen Sie die folgenden Schritte aus, um einen Bewertungsbericht zur Datenbankmigration zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Datenbankmigration

1. Stellen Sie sicher, dass Sie eine Zuordnungsregel für das Quelldatenbankschema erstellt haben, für das Sie einen Bewertungsbericht erstellen möchten. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.

2. Wählen Sie im Menü Ansicht die Option Hauptansicht.
3. Wählen Sie im linken Bereich, in dem Ihr Quelldatenbankschema angezeigt wird, ein Schemaobjekt aus, für das Sie einen Bewertungsbericht erstellen möchten. Um mehrere Datenbankschemas in den Bericht aufzunehmen, wählen Sie den übergeordneten Knoten aus, zum Beispiel Schemas.

   Stellen Sie sicher, dass Sie die Kontrollkästchen für alle Schemaobjekte aktiviert haben, für die Sie einen Bewertungsbericht erstellen möchten.
4. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie dann Bericht erstellen aus.



Einen Bewertungsbericht anzeigen in AWS Schema Conversion Tool

Nach der Erstellung des Bewertungsberichts wird die entsprechende Ansicht mit folgenden Registerkarten geöffnet:

- Übersicht
- Action Items (Aktionselemente)

Die Registerkarte Summary (Zusammenfassung) enthält Elemente, die automatisch konvertiert oder nicht konvertiert wurden.

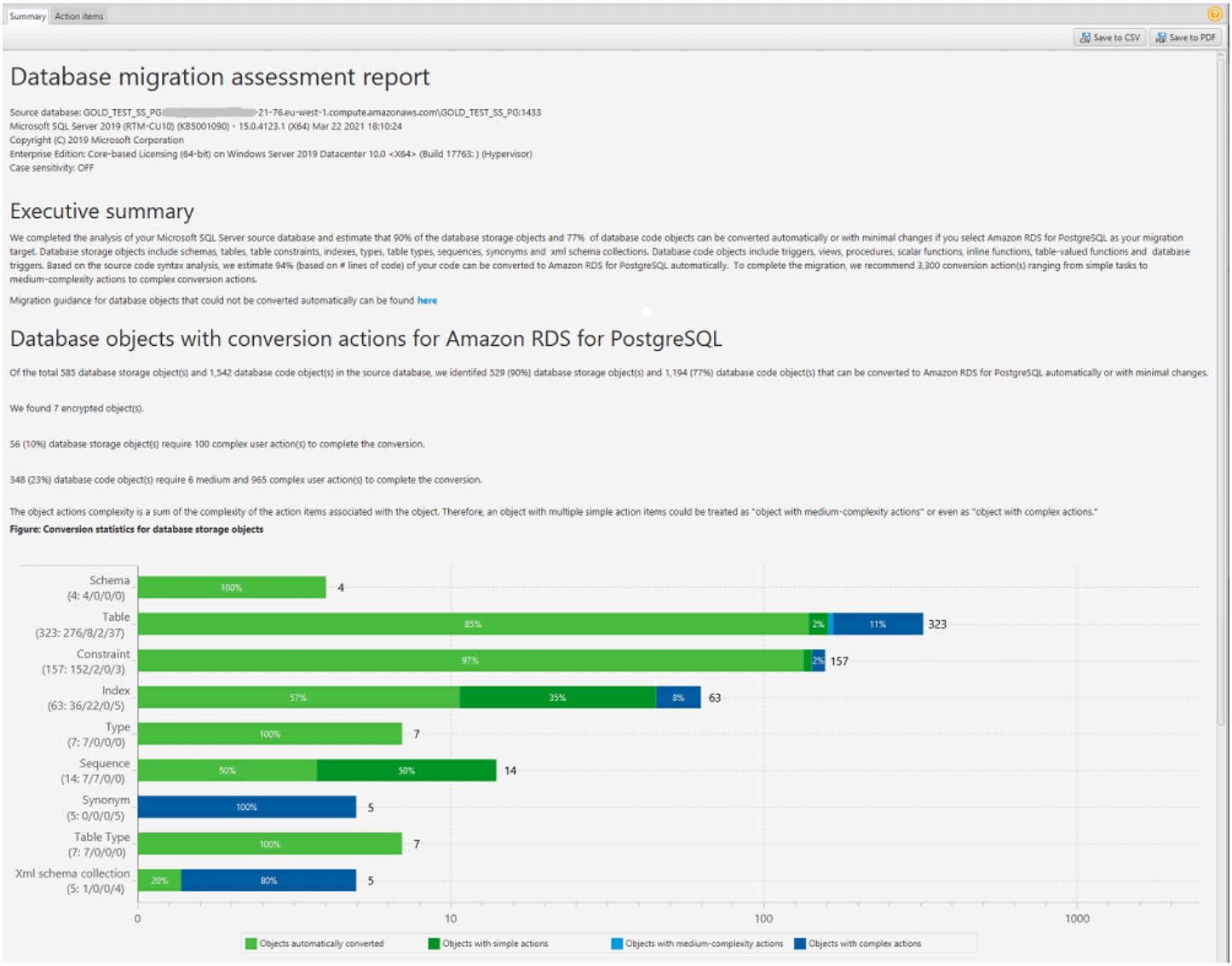
Die Registerkarte Action Items (Aktionselemente) zeigt Elemente an, die nicht automatisch konvertiert werden konnten, sowie Empfehlungen zur Handhabung dieser Elemente.

Themen

- Zusammenfassung des Bewertungsberichts
- Aktionselemente im Bewertungsbericht
- Warnmeldung im Bewertungsbericht

Zusammenfassung des Bewertungsberichts

Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es werden sowohl die automatisch konvertierten Elemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.

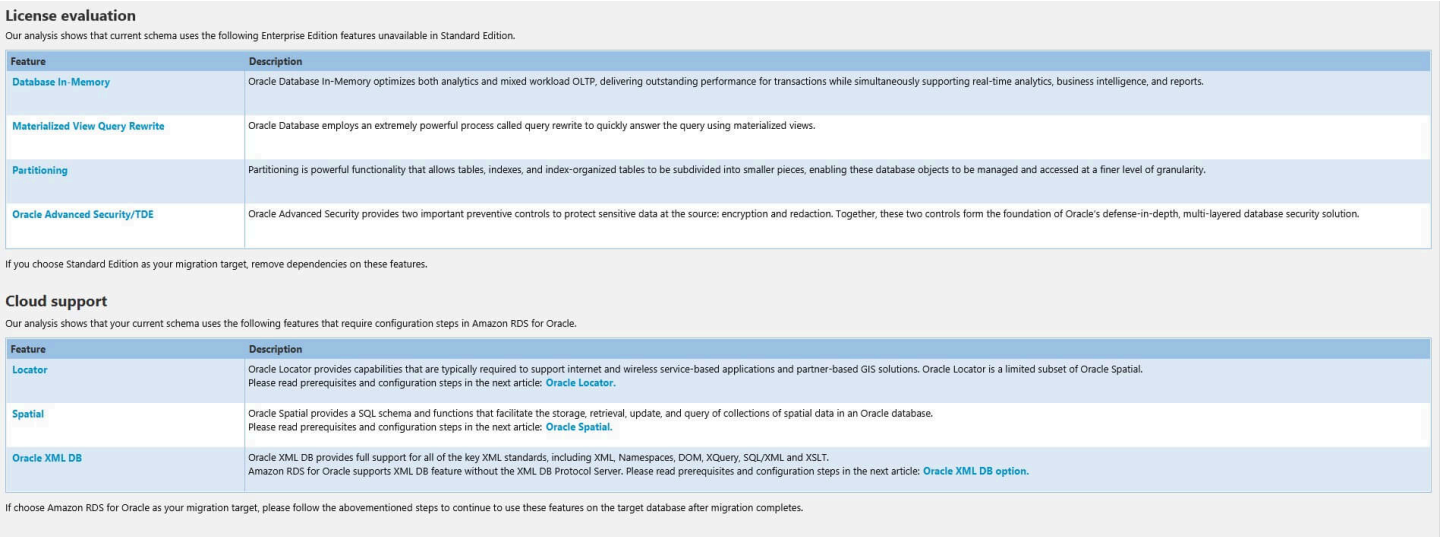


Für die Schemaelemente, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden können, enthält die Zusammenfassung eine Aufwandsschätzung zur Erstellung von Schemaelementen in Ihrer Ziel-DB-Instance, die mit denen in der Quelle gleichgesetzt werden können.

Im Bericht wird der geschätzte Zeitaufwand für die Konvertierung dieser Schemaelemente wie folgt kategorisiert:

- Einfach — Aktionen, die in weniger als zwei Stunden abgeschlossen werden können.
- Mittel — Aktionen, die komplexer sind und in zwei bis sechs Stunden abgeschlossen werden können.
- Signifikant — Aktionen, die sehr komplex sind und mehr als sechs Stunden in Anspruch nehmen.

Der Abschnitt Lizenzbewertung und Cloud-Support enthält Informationen zum Verschieben Ihres vorhandenen lokalen Datenbankschemas auf eine Amazon RDS-DB-Instance, auf der dieselbe Engine ausgeführt wird. Wenn Sie beispielsweise Lizenztypen ändern möchten, finden Sie in diesem Abschnitt Informationen dazu, welche Funktionen Ihrer aktuellen Datenbank entfernt werden sollten.



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.

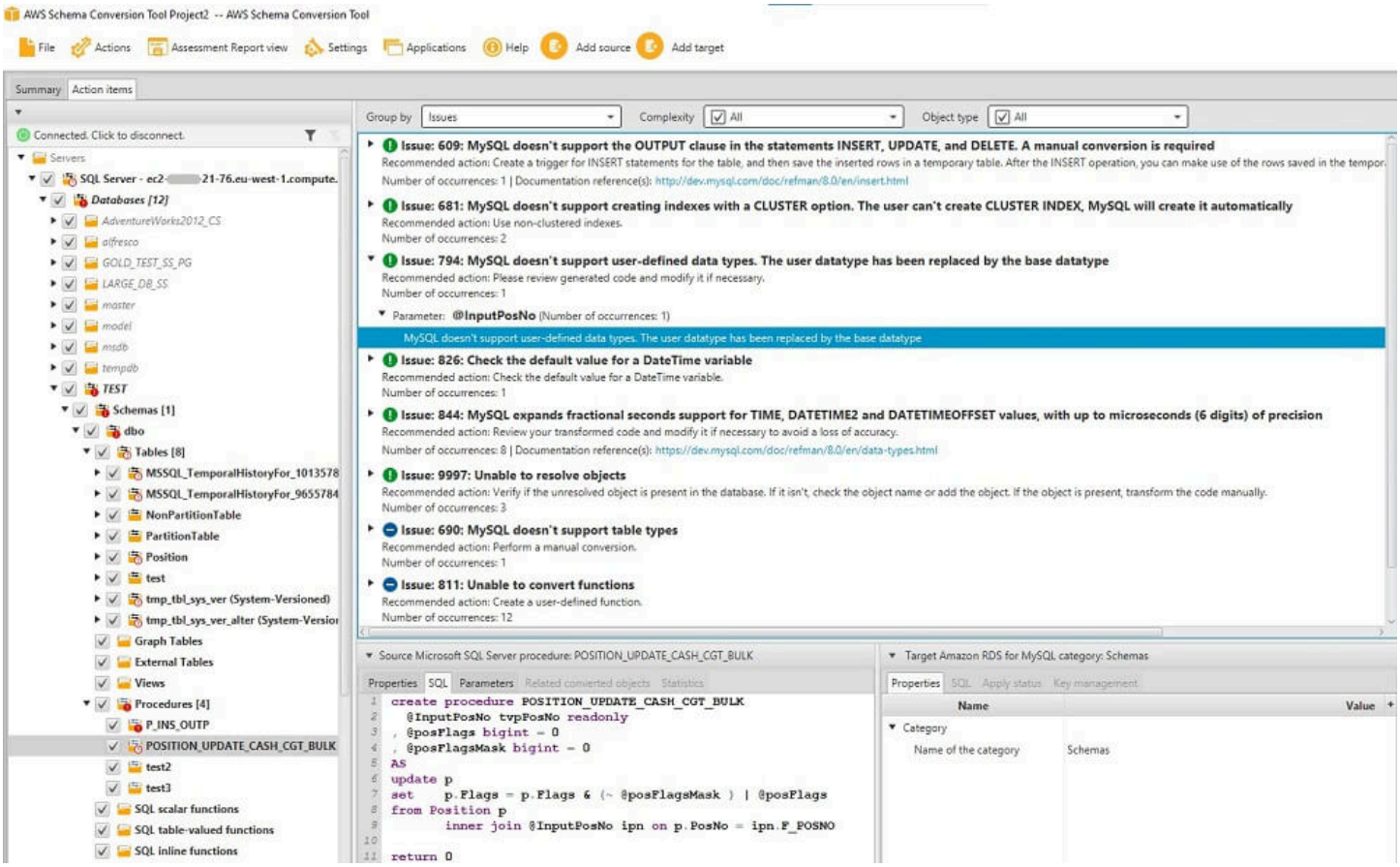
Feature	Description
Locator	Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. Please read prerequisites and configuration steps in the next article: Oracle Locator.
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: Oracle Spatial.
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: Oracle XML DB option.

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.

Aktionselemente im Bewertungsbericht

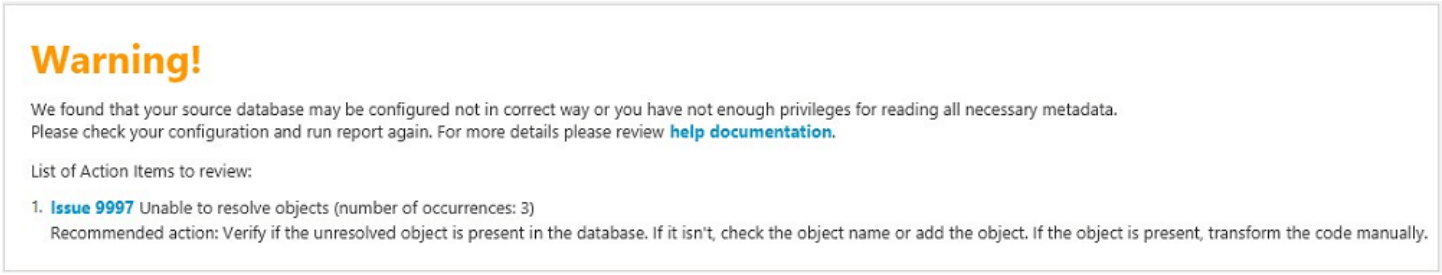
Im Bewertungsbericht ist auch eine Registerkarte Action Items vorhanden. Diese Registerkarte enthält eine Liste von Elementen, die nicht automatisch in die Datenbank-Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Wenn Sie ein Aktionselement aus der Liste auswählen, markiert AWS SCT das entsprechende Element im Schema, für das diese Aktion gilt.

Der Bericht enthält zudem Empfehlungen für die manuelle Konvertierung des Schemaelements. Nach der Ausführung der Bewertung zeigen Ihnen beispielsweise detaillierte Berichte für die Datenbank/das Schema, wie viel Aufwand erforderlich ist, um die Empfehlungen für die Konvertierung von Aktionselementen zu entwerfen und umzusetzen. Weitere Informationen zum Festlegen der Handhabung von manuellen Konvertierungen finden Sie unter Schemas mit AWS SCT konvertieren.



Warnmeldung im Bewertungsbericht

Um die Komplexität der Konvertierung in eine andere Datenbank-Engine beurteilen zu können, AWS SCT ist Zugriff auf Objekte in Ihrer Quelldatenbank erforderlich. Wenn SCT keine Bewertung durchführen kann, weil beim Scannen Probleme aufgetreten sind, wird eine Warnmeldung ausgegeben, die darauf hinweist, dass der Gesamtkonvertierungsprozentsatz reduziert wurde.



Im Folgenden sind die Gründe aufgeführt AWS SCT , aus denen beim Scannen Probleme auftreten können:

- Das mit der Datenbank verbundene Benutzerkonto hat nicht Zugriff auf alle benötigten Objekte.
- Ein im Schema zitiertes Objekt ist in der Datenbank nicht mehr vorhanden.

- SCT versucht, ein verschlüsseltes Objekt zu bewerten.

Weitere Informationen zu den von SCT benötigten Sicherheitsberechtigungen und -berechtigungen für Ihre Datenbank finden Sie im entsprechenden Abschnitt Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool zur Quelldatenbank in diesem Handbuch.

Speichern Sie den Bewertungsbericht in AWS Schema Conversion Tool

Nachdem Sie einen Bewertungsbericht zur Datenbankmigration erstellt haben, können Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV) speichern.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als PDF-Datei

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“ aus.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts „Als PDF speichern“.

Um einen Bewertungsbericht zur Datenbankmigration als CSV-Datei zu speichern

1. Wählen Sie im oberen Menü Ansicht und dann Ansicht des Bewertungsberichts aus.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts die Option Als CSV speichern aus.

Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

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 identifed 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.

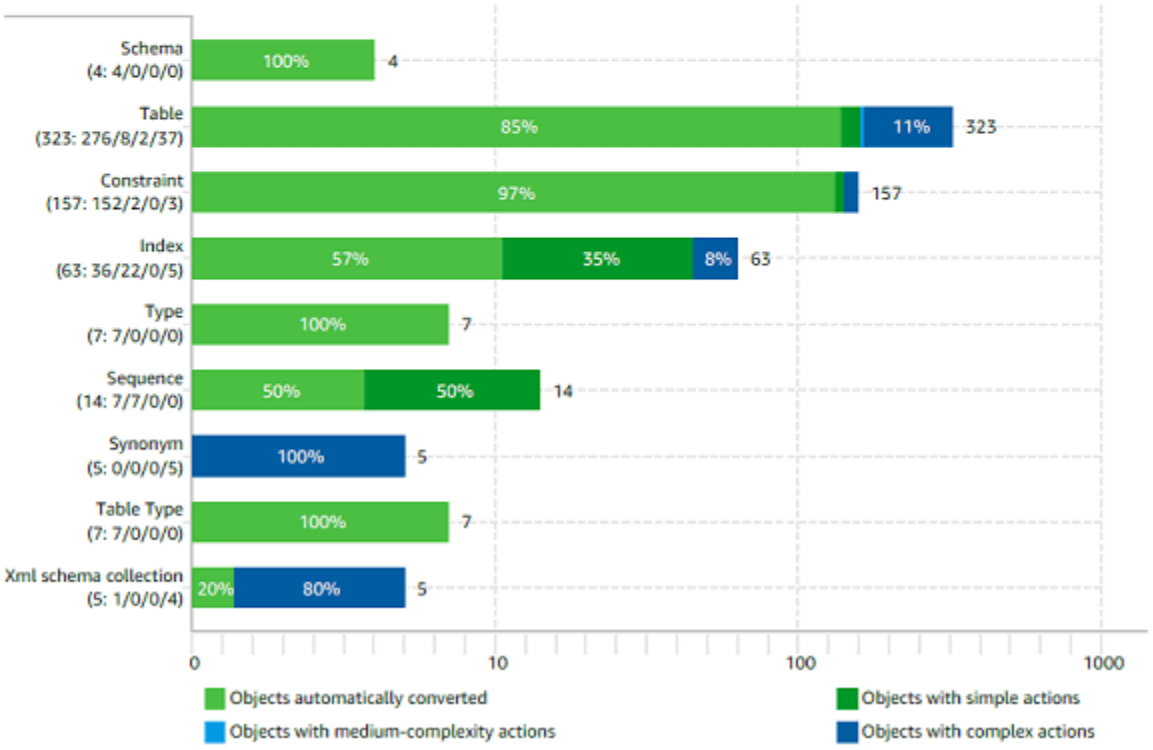
We found 7 encrypted object(s).

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



Wenn Sie die Option In CSV speichern wählen, werden drei CSV-Dateien AWS SCT erstellt.

Die erste CSV-Datei enthält die folgenden Informationen über die Aktionselemente:

- Kategorie
- Vorkommen — Der Dateiname, die Zeilennummer und die Position des Elements
- Nummer des Aktionspunkts
- Betreff
- Gruppe

- Beschreibung
- Dokumentationsreferenzen
- Empfohlene Aktion
- Geschätzte Komplexität

Die zweite CSV-Datei enthält das `Action_Items_Summary` Suffix in ihrem Namen und enthält Informationen über die Anzahl der Vorkommen aller Aktionspunkte.

Im folgenden Beispiel geben die Werte in der Spalte Aufwand der Lernkurve den Aufwand an, der erforderlich ist, um einen Ansatz für die Konvertierung der einzelnen Aktionspunkte zu entwickeln. Die Werte in der Spalte Aufwand zur Konvertierung eines Vorkommens des Aktionspunkts geben den Aufwand an, der erforderlich ist, um die einzelnen Aktionspunkte entsprechend dem entworfenen Ansatz zu konvertieren. Die zur Angabe des erforderlichen Aufwandes verwendeten Werte basieren auf einer gewichteten Skala, die von niedrig (am wenigsten) bis hoch (am meisten) reicht.

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

Die dritte CSV-Datei enthält `Summary` in ihrem Namen die folgende Zusammenfassung:

- Kategorie
- Anzahl der Objekte
- Automatisch konvertierte Objekte
- Objekte mit einfachen Aktionen
- Objekte mit Aktionen mittlerer Komplexität
- Objekte mit komplexen Aktionen
- Gesamtzahl der Codezeilen

Konfiguration eines Bewertungsberichts in AWS Schema Conversion Tool

Sie können den Umfang der Details konfigurieren, die in den Bewertungsberichten AWS SCT enthalten sein sollen.

So konfigurieren Sie einen Bewertungsbericht zur Datenbankmigration

1. Wählen Sie im Menü Einstellungen die Option Allgemeine Einstellungen und dann Bewertungsbericht aus.
2. Wählen Sie für Vorkommen von Aktionspunkten die Option Nur die ersten fünf Probleme aus, um die Anzahl der Aktionspunkte eines einzigen Typs im Bewertungsbericht zu begrenzen. Wählen Sie Alle Probleme aus, um alle Aktionspunkte jedes Typs in den Bewertungsbericht aufzunehmen.
3. Wählen Sie für analysierte SQL-Skriptdateien die Option Nicht mehr als ***X*** Dateien auflisten, um die Anzahl der SQL-Skriptdateien im Bewertungsbericht auf zu beschränken*X*. Geben Sie die Anzahl der Dateien ein. Wählen Sie Alle analysierten Dateien auflisten, um alle SQL-Skriptdateien in den Bewertungsbericht aufzunehmen.
4. Wählen Sie Berichte nach dem Speichern öffnen, um die Datei automatisch zu öffnen, nachdem Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration gespeichert haben. Weitere Informationen finden Sie unter

Nachdem Sie einen Bewertungsbericht zur Datenbankmigration erstellt haben, können Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV) speichern.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als PDF-Datei

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“ aus.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts „Als PDF speichern“.

Um einen Bewertungsbericht zur Datenbankmigration als CSV-Datei zu speichern

1. Wählen Sie im oberen Menü Ansicht und dann Ansicht des Bewertungsberichts aus.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts die Option Als CSV speichern aus.

Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

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 identifed 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.

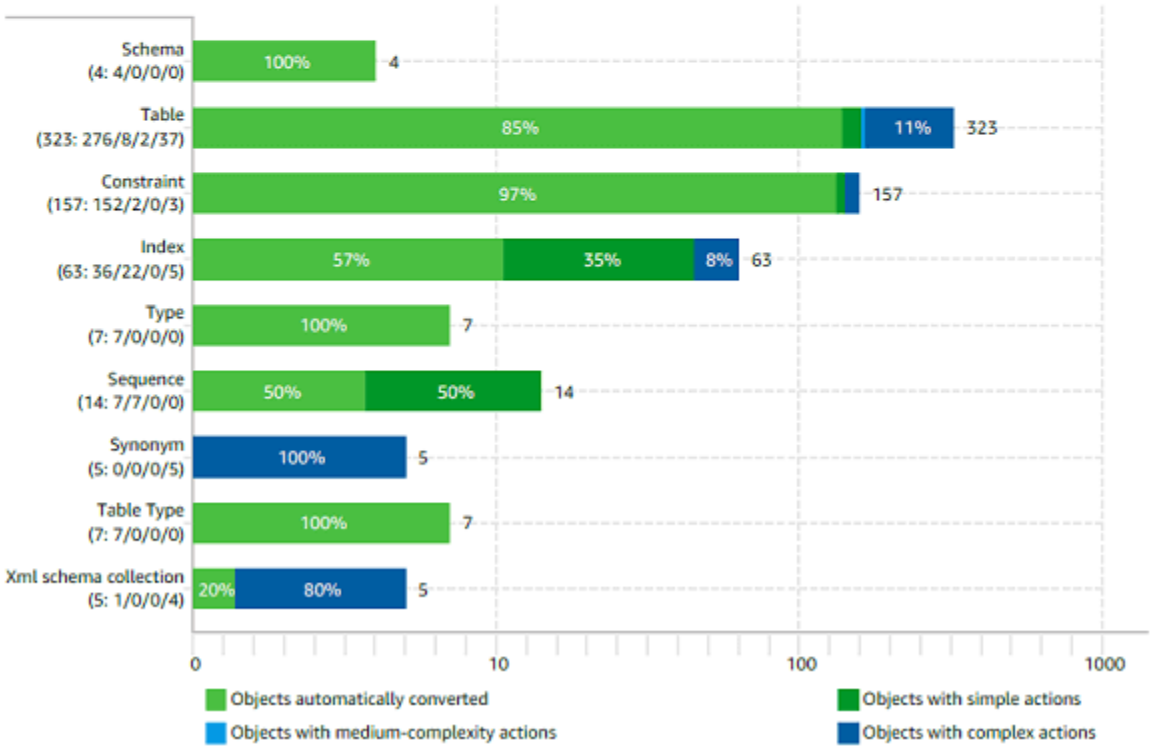
We found 7 encrypted object(s).

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



Wenn Sie die Option In CSV speichern wählen, werden drei CSV-Dateien AWS SCT erstellt.

Die erste CSV-Datei enthält die folgenden Informationen über die Aktionselemente:

- Kategorie
- Vorkommen — Der Dateiname, die Zeilennummer und die Position des Elements

- Nummer des Aktionspunkts

- Betreff
- Gruppe
- Beschreibung
- Dokumentationsreferenzen
- Empfohlene Aktion
- Geschätzte Komplexität

Die zweite CSV-Datei enthält das `Action_Items_Summary` Suffix in ihrem Namen und enthält Informationen über die Anzahl der Vorkommen aller Aktionspunkte.

Im folgenden Beispiel geben die Werte in der Spalte Aufwand der Lernkurve den Aufwand an, der erforderlich ist, um einen Ansatz für die Konvertierung der einzelnen Aktionspunkte zu entwickeln. Die Werte in der Spalte Aufwand zur Konvertierung eines Vorkommens des Aktionspunkts geben den Aufwand an, der erforderlich ist, um die einzelnen Aktionspunkte entsprechend dem entworfenen Ansatz zu konvertieren. Die zur Angabe des erforderlichen Aufwandes verwendeten Werte basieren auf einer gewichteten Skala, die von niedrig (am wenigsten) bis hoch (am meisten) reicht.

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

Die dritte CSV-Datei enthält `Summary` in ihrem Namen die folgende Zusammenfassung:

- Kategorie
- Anzahl der Objekte
- Automatisch konvertierte Objekte
- Objekte mit einfachen Aktionen
- Objekte mit Aktionen mittlerer Komplexität
- Objekte mit komplexen Aktionen
- Gesamtzahl der Codezeilen

.

Einen Bewertungsbericht für mehrere Server erstellen in AWS Schema Conversion Tool

Erstellen Sie einen Bewertungsbericht für mehrere Server, um die beste Zielrichtung für Ihre gesamte Umgebung zu ermitteln.

In einem Bewertungsbericht für mehrere Server werden mehrere Server anhand der Eingaben bewertet, die Sie für jede Schemadefinition angeben, die Sie bewerten möchten. Ihre Schemadefinition enthält Verbindungsparameter für den Datenbankserver und den vollständigen Namen jedes Schemas. Erstellt nach der Bewertung jedes AWS SCT Schemas einen zusammenfassenden, aggregierten Bewertungsbericht für die Datenbankmigration zwischen Ihren verschiedenen Servern. Dieser Bericht zeigt die geschätzte Komplexität für jedes mögliche Migrationsziel.

Sie können ihn verwenden AWS SCT , um einen Bewertungsbericht für mehrere Server für die folgenden Quell- und Zieldatenbanken zu erstellen.

Quelldatenbank	Zieldatenbank
Amazon Redshift	Amazon Redshift
Azure SQL Database	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Azure Synapse-Analytik	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM Db2 für z/OS	Amazon Aurora MySQL-kompatible Edition (Aurora MySQL), Amazon Aurora PostgreSQ L-kompatible Edition (Aurora PostgreSQL), MySQL, PostgreSQL

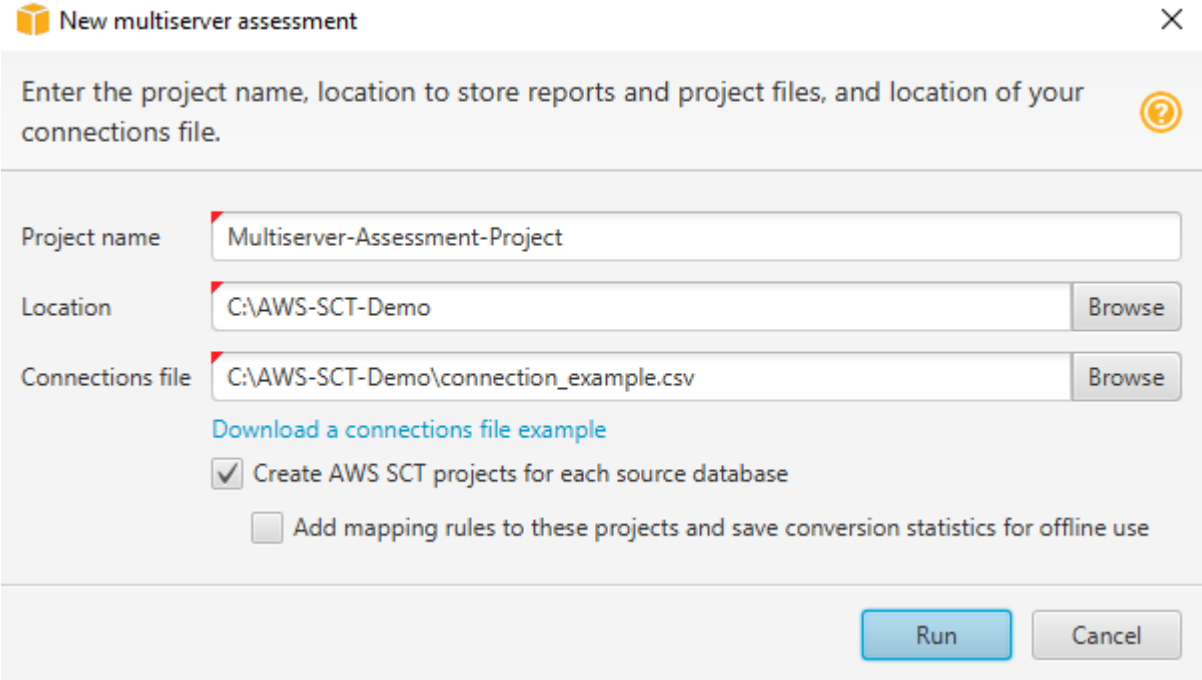
Quelldatenbank	Zieldatenbank
IBM Db2 (LUW)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, Babelfish für Aurora PostgreSQ L, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQ L
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

Durchführung einer Multiserver-Assessment

Gehen Sie wie folgt vor, um eine Multiserverbewertung mit durchzuführen. AWS SCT Sie müssen kein neues Projekt erstellen, AWS SCT um eine Multiserverbewertung durchzuführen. Bevor Sie beginnen, stellen Sie sicher, dass Sie eine Datei mit kommagetrennten Werten (CSV) mit Datenbankverbindungsparametern vorbereitet haben. Stellen Sie außerdem sicher, dass Sie alle erforderlichen Datenbanktreiber installiert haben, und legen Sie den Speicherort der Treiber in den AWS SCT Einstellungen fest. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

Um eine Multiserver-Bewertung durchzuführen und einen zusammengefassten Zusammenfassungsbericht zu erstellen

1. Wählen Sie AWS SCT unter Datei, Neue Multiserverbewertung aus. Das Dialogfeld Neue Multiserver-Bewertung wird geöffnet.



2. Wählen Sie Beispiel für eine Verbindungsdatei herunterladen, um eine leere Vorlage einer CSV-Datei mit Datenbankverbindungsparametern herunterzuladen.
3. Geben Sie Werte für Projektname, Speicherort (zum Speichern von Berichten) und Verbindungsdatei (eine CSV-Datei) ein.
4. Wählen Sie AWS SCT Projekte für jede Quelldatenbank erstellen, um nach der Erstellung des Bewertungsberichts automatisch Migrationsprojekte zu erstellen.
5. Wenn die Option AWS SCT Projekte für jede Quelldatenbank erstellen aktiviert ist, können Sie Zuordnungsregeln zu diesen Projekten hinzufügen auswählen und Konvertierungsstatistiken für die Offline-Verwendung speichern. In diesem Fall AWS SCT fügt jedem Projekt Zuordnungsregeln hinzu und speichert die Metadaten der Quelldatenbank im Projekt. Weitere Informationen finden Sie unter Verwenden Sie den Offline-Modus in AWS Schema Conversion Tool.
6. Wählen Sie Ausführen aus.

   Es erscheint ein Fortschrittsbalken, der das Tempo der Datenbankbewertung anzeigt. Die Anzahl der Ziel-Engines kann sich auf die Laufzeit der Bewertung auswirken.

7. Wählen Sie Ja, wenn die folgende Meldung angezeigt wird: Die vollständige Analyse aller Datenbankserver kann einige Zeit dauern. Möchten Sie fortfahren?

   Wenn der Multiserver-Assessment-Bericht fertig ist, wird ein entsprechender Bildschirm angezeigt.
8. Wählen Sie Bericht öffnen, um den zusammengefassten Bewertungsbericht anzuzeigen.

AWS SCT Generiert standardmäßig einen aggregierten Bericht für alle Quelldatenbanken und einen detaillierten Bewertungsbericht für jeden Schemanamen in einer Quelldatenbank. Weitere Informationen finden Sie unter Suchen und Anzeigen von Berichten.

Wenn die Option AWS SCT Projekte für jede Quelldatenbank erstellen aktiviert ist, AWS SCT wird für jede Quelldatenbank ein leeres Projekt erstellt. AWS SCT erstellt auch Bewertungsberichte, wie zuvor beschrieben. Nachdem Sie diese Bewertungsberichte analysiert und das Migrationsziel für jede Quelldatenbank ausgewählt haben, fügen Sie Zieldatenbanken zu diesen leeren Projekten hinzu.

Wenn die Option Zuordnungsregeln zu diesen Projekten hinzufügen und Konvertierungsstatistiken für die Offline-Verwendung speichern aktiviert ist, AWS SCT wird für jede Quelldatenbank ein Projekt erstellt. Diese Projekte enthalten die folgenden Informationen:

- Ihre Quelldatenbank und eine virtuelle Zieldatenbankplattform. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
- Eine Zuordnungsregel für dieses Quell-Ziel-Paar. Weitere Informationen finden Sie unter Zuordnung von Datentypen.
- Ein Bewertungsbericht zur Datenbankmigration für dieses Quell-Ziel-Paar.
- Metadaten des Quellschemas, mit denen Sie dieses AWS SCT Projekt im Offline-Modus verwenden können. Weitere Informationen finden Sie unter Verwenden Sie den Offline-Modus in AWS Schema Conversion Tool.

Eine CSV-Eingabedatei wird vorbereitet

Um Verbindungsparameter als Eingabe für den Multiserver-Assessmentbericht bereitzustellen, verwenden Sie eine CSV-Datei, wie im folgenden Beispiel gezeigt.

```
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
```

Im vorherigen Beispiel werden die beiden Schemanamen für die Datenbank durch ein Semikolon getrennt. `Sales` Außerdem werden die beiden Zieldatenbank-Migrationsplattformen für die Datenbank durch ein Semikolon voneinander getrennt. `Sales`

Außerdem verwendet das vorherige Beispiel die Verbindung AWS Secrets Manager zur `Customers` Datenbank und die Windows-Authentifizierung, um eine Verbindung zur Datenbank herzustellen. `HR`

Sie können eine neue CSV-Datei erstellen oder eine Vorlage für eine CSV-Datei von herunterladen AWS SCT und die erforderlichen Informationen eingeben. Stellen Sie sicher, dass die erste Zeile Ihrer CSV-Datei dieselben Spaltennamen wie im vorherigen Beispiel enthält.

Um eine Vorlage der CSV-Eingabedatei herunterzuladen

1. Fangen Sie an AWS SCT.
2. Wählen Sie „Datei“ und anschließend „Neues Multiserver-Assessment“.
3. Wählen Sie „Beispiel für eine Verbindungsdatei herunterladen“ aus.

Stellen Sie sicher, dass Ihre CSV-Datei die folgenden Werte enthält, die von der Vorlage bereitgestellt werden:

- Name — Die Textbeschriftung, mit der Sie Ihre Datenbank identifizieren können. AWS SCT zeigt dieses Textlabel im Bewertungsbericht an.
- Beschreibung — Ein optionaler Wert, mit dem Sie zusätzliche Informationen zur Datenbank angeben können.
- Secret Manager Key — Der Name des Secrets, das Ihre Datenbankanmeldedaten speichert, in der AWS Secrets Manager. Um Secrets Manager zu verwenden, stellen Sie sicher, dass Sie AWS Profile in speichern AWS SCT. Weitere Informationen finden Sie unter Konfiguration AWS Secrets Manager in der AWS Schema Conversion Tool.

> **Important**
>
> AWS SCT ignoriert den Secret Manager-Schlüsselparameter, wenn Sie die Server-IP -, Port -, Anmelde - und Kennwortparameter in die Eingabedatei aufnehmen.

- Server-IP — Der DNS-Name (Domain Name Service) oder die IP-Adresse Ihres Quelldatenbankservers.
- Port — Der Port, der für die Verbindung zu Ihrem Quelldatenbankserver verwendet wird.
- Dienstname — Wenn Sie einen Dienstnamen verwenden, um eine Verbindung zu Ihrer Oracle-Datenbank herzustellen, der Name des Oracle-Dienstes, zu dem die Verbindung hergestellt werden soll.
- Datenbankname — Der Datenbankname. Verwenden Sie für Oracle-Datenbanken die Oracle-System-ID (SID).
- BigQuery Pfad — der Pfad zur Dienstkonto-Schlüsseldatei für Ihre BigQuery Quelldatenbank. Weitere Hinweise zum Erstellen dieser Datei finden Sie unterRechte für BigQuery als Quelle.
- Source Engine — Der Typ Ihrer Quelldatenbank. Verwenden Sie einen der folgenden Werte:
  - AZURE_MSSQL für eine Azure SQL-Datenbank.
  - AZURE_SYNAPSE für eine Azure Synapse Analytics-Datenbank.
  - GOOGLE_BIGQUERY für eine Datenbank. BigQuery
  - DB2ZOS für eine IBM Db2 for z/OS-Datenbank.
  - DB2LUW für eine IBM Db2 LUW-Datenbank.
  - GREENPLUM für eine Greenplum-Datenbank.
  - MSSQL für eine Microsoft SQL Server-Datenbank.
  - MYSQL für eine MySQL-Datenbank.
  - NETEZZA für eine Netezza-Datenbank.
  - ORACLE für eine Oracle-Datenbank.
  - POSTGRESQL für eine PostgreSQL-Datenbank.
  - REDSHIFT für eine Amazon Redshift Redshift-Datenbank.
  - SNOWFLAKE für eine Snowflake-Datenbank.
  - SYBASE_ASE für eine SAP ASE-Datenbank.
  - TERADATA für eine Teradata-Datenbank.
  - VERTICA für eine Vertica-Datenbank.

- Schemanamen — Die Namen der Datenbankschemas, die in den Bewertungsbericht aufgenommen werden sollen.

  Verwenden Sie für Azure SQL Database, Azure Synapse Analytics BigQuery, Netezza, SAP ASE, Snowflake und SQL Server das folgende Format des Schemanamens:

  db_name.*schema_name*

  Ersetzen Sie es durch *db_name* den Namen der Quelldatenbank.

  *schema_name*Ersetzen Sie durch den Namen des Quellschemas.

  Schließen Sie Datenbank- oder Schemanamen ein, die einen Punkt in doppelten Anführungszeichen enthalten, wie folgt:`"database.name"."schema.name"`.

  Trennen Sie mehrere Schemanamen, indem Sie Semikolons verwenden, wie folgt:. `Schema1;Schema2`

  Bei den Datenbank- und Schemanamen wird zwischen Groß- und Kleinschreibung unterschieden.

  Verwenden Sie Prozent (`%`) als Platzhalter, um eine beliebige Anzahl von Symbolen im Datenbank- oder Schemanamen zu ersetzen. Im vorherigen Beispiel wird die Prozentzahl (`%`) als Platzhalter verwendet, um alle Schemas aus der `employees` Datenbank in den Bewertungsbericht aufzunehmen.
- Windows-Authentifizierung verwenden — Wenn Sie die Windows-Authentifizierung verwenden, um eine Verbindung zu Ihrer Microsoft SQL Server-Datenbank herzustellen, geben Sie true ein. Weitere Informationen finden Sie unter Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle.
- Login — Der Benutzername für die Verbindung mit Ihrem Quelldatenbankserver.
- Passwort — Das Passwort für die Verbindung zu Ihrem Quelldatenbankserver.
- SSL verwenden — Wenn Sie Secure Sockets Layer (SSL) verwenden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, geben Sie true ein.
- Trust Store — Der Trust Store, der für Ihre SSL-Verbindung verwendet werden soll.
- Schlüsselspeicher — Der Schlüsselspeicher, den Sie für Ihre SSL-Verbindung verwenden möchten.
- SSL-Authentifizierung — Wenn Sie die SSL-Authentifizierung per Zertifikat verwenden, geben Sie true ein.

- Target Engines — Die Zieldatenbankplattformen. Verwenden Sie die folgenden Werte, um ein oder mehrere Ziele im Bewertungsbericht anzugeben:
  - AURORA_MYSQL für eine Aurora MySQL-kompatible Datenbank.
  - AURORA_POSTGRESQL für eine Aurora PostgreSQL-kompatible Datenbank.
  - BABELFISH für eine Babelfish for Aurora PostgreSQL-Datenbank.
  - MARIA_DB für eine MariaDB-Datenbank.
  - MSSQL für eine Microsoft SQL Server-Datenbank.
  - MYSQL für eine MySQL-Datenbank.
  - ORACLE für eine Oracle-Datenbank.
  - POSTGRESQL für eine PostgreSQL-Datenbank.
  - REDSHIFT für eine Amazon Redshift Redshift-Datenbank.

  Trennen Sie mehrere Ziele, indem Sie Semikolons wie folgt verwenden:. `MYSQL;MARIA_DB` Die Anzahl der Ziele wirkt sich auf die Zeit aus, die für die Durchführung der Bewertung benötigt wird.

Suchen und Anzeigen von Berichten

Bei der Multiserver-Assessment werden zwei Arten von Berichten generiert:

- Ein aggregierter Bericht aller Quelldatenbanken.
- Ein detaillierter Bewertungsbericht der Zieldatenbanken für jeden Schemanamen in einer Quelldatenbank.

Berichte werden in dem Verzeichnis gespeichert, das Sie im Dialogfeld Neue Multiserver-Bewertung für Speicherort ausgewählt haben.

Um auf die detaillierten Berichte zuzugreifen, können Sie in den Unterverzeichnissen navigieren, die nach Quelldatenbank, Schemaname und Zieldatenbank-Engine organisiert sind.

Aggregierte Berichte enthalten Informationen zur Konvertierungskomplexität einer Zieldatenbank in vier Spalten. Die Spalten enthalten Informationen zur Konvertierung von Codeobjekten, Speicherobjekten, Syntaxelementen und zur Komplexität der Konvertierung.

Das folgende Beispiel zeigt Informationen zur Konvertierung von zwei Oracle-Datenbankschemas nach 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

Dieselben vier Spalten werden an die Berichte für jede weitere angegebene Zieldatenbank-Engine angehängt.

Einzelheiten zum Lesen dieser Informationen finden Sie im Folgenden.

Ergebnis für einen aggregierten Bewertungsbericht

Der aggregierte Bewertungsbericht zur Migration von Multiserver-Datenbanken in AWS Schema Conversion Tool ist eine CSV-Datei mit den folgenden Spalten:

- `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`*

Um Informationen zu sammeln, werden vollständige Bewertungsberichte AWS SCT ausgeführt und anschließend die Berichte nach Schemas zusammengefasst.

In dem Bericht zeigen die folgenden drei Felder den Prozentsatz der möglichen automatischen Konversionen auf der Grundlage der Bewertung:

Konvertierung von Codeobjekten (%)

> Der Prozentsatz der Codeobjekte im Schema, die automatisch oder mit minimalen Änderungen konvertiert werden AWS SCT können. Zu den Codeobjekten gehören Prozeduren, Funktionen, Ansichten und ähnliches.

Konvertierung von Speicherobjekten%

Der Prozentsatz der Speicherobjekte, die SCT automatisch oder mit minimalen Änderungen konvertieren kann. Zu den Speicherobjekten gehören Tabellen, Indizes, Einschränkungen und ähnliches.

Konvertierung von Syntaxelementen%

Der Prozentsatz der Syntaxelemente, die SCT automatisch konvertieren kann. Zu den Syntaxelementen gehören `SELECTFROM`,`DELETE`,, und `JOIN` Klauseln und ähnliches.

Die Berechnung der Konvertierungskomplexität basiert auf dem Konzept der Aktionspunkte. Ein Aktionspunkt spiegelt eine Art von Problem wider, das im Quellcode gefunden wurde und das Sie während der Migration zu einem bestimmten Ziel manuell beheben müssen. Ein Aktionspunkt kann mehrfach vorkommen.

Eine gewichtete Skala gibt den Grad der Komplexität an, mit dem eine Migration durchgeführt werden muss. Die Zahl 1 steht für die niedrigste und die Zahl 10 für die höchste Komplexitätsstufe.

Datenbankschemas konvertieren in AWS Schema Conversion Tool

Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um Ihre vorhandenen Datenbankschemas von einer Datenbank-Engine in eine andere zu konvertieren. Das Konvertieren einer Datenbank mithilfe der AWS SCT Benutzeroberfläche kann recht einfach sein, aber es gibt mehrere Dinge, die Sie beachten sollten, bevor Sie die Konvertierung durchführen.

Sie können z. B. Folgendes verwenden AWS SCT :

- Sie können AWS SCT es verwenden, um ein vorhandenes lokales Datenbankschema in eine Amazon RDS-DB-Instance zu kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind.
- In einigen Fällen können Datenbankfunktionen nicht in äquivalente Amazon RDS-Funktionen konvertiert werden. Wenn Sie eine Datenbank auf der Amazon Elastic Compute Cloud (Amazon EC2) -Plattform hosten und selbst verwalten, können Sie diese Funktionen emulieren, indem Sie sie durch Dienste ersetzen AWS .
- AWS SCT automatisiert einen Großteil des Prozesses der Konvertierung Ihres OLTP-Datenbankschemas (Online Transaction Processing) in eine MySQL-DB-Instance von Amazon Relational Database Service (Amazon RDS), einen Amazon Aurora Aurora-DB-Cluster oder eine PostgreSQL-DB-Instance. Die Quell- und Zieldatenbank-Engines enthalten viele verschiedene Funktionen und Fähigkeiten und AWS SCT versuchen, wo immer möglich, ein gleichwertiges Schema in Ihrer Amazon RDS-DB-Instance zu erstellen. Wenn keine direkte Konvertierung möglich ist, finden AWS SCT Sie hier eine Liste möglicher Aktionen, die Sie ergreifen können.

Themen

- Anwenden von Migrationsregeln in AWS Schema Conversion Tool
- Schemas mit AWS SCT konvertieren
- Manuelles Konvertieren von Schemas in AWS SCT
- Aktualisierung und Aktualisierung der konvertierten Schemas in AWS SCT
- Speichern und Anwenden konvertierter Schemas in AWS SCT
- Schemas vergleichen in AWS Schema Conversion Tool
- Anzeige verwandter transformierter Objekte in AWS Schema Conversion Tool

AWS SCT unterstützt die folgenden OLTP-Konvertierungen (Online Transaction Processing).

Quelldatenbank	Zieldatenbank
IBM Db2 für z/OS (Version 12)	Amazon Aurora MySQL-kompatible Edition, Amazon Aurora PostgreSQL-kompatible Ausgabe, MySQL, PostgreSQL
IBM Db2 LUW (Versionen 9.1, 9.5, 9.7, 10.5, 11.1 und 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft Azure SQL Database	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (Version 2008 R2 und höher)	Aurora MySQL, Aurora PostgreSQL, Babelfish für Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (Version 5.5 und höher)	Aurora PostgreSQL, MySQL, PostgreSQL Sie können Schema und Daten von MySQL zu einem Aurora MySQL-DB-Cluster migrieren , ohne es zu verwenden AWS SCT. Weitere Informationen finden Sie unter Daten in einen Amazon Aurora Aurora-DB-Cluster migrieren.
Oracle (Version 10.2 und höher)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (Version 9.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12,5, 15,0, 15,5, 15,7 und 16,0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Hinweise zur Konvertierung eines Data Warehouse-Schemas finden Sie unter. Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT

Um Ihr Datenbankschema in Amazon RDS zu konvertieren, führen Sie die folgenden allgemeinen Schritte aus:

- Migrationsregeln erstellen in AWS SCT— Bevor Sie Ihr Schema mit konvertieren AWS SCT, können Sie Regeln einrichten, die den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern.
- ???— AWS SCT erstellt eine lokale Version des konvertierten Schemas, die Sie überprüfen können, wendet sie jedoch erst auf Ihre Ziel-DB-Instance an, wenn Sie bereit sind.
- Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool— AWS SCT erstellt einen Bewertungsbericht zur Datenbankmigration, in dem die Schemaelemente detailliert beschrieben werden, die nicht automatisch konvertiert werden können. Sie können diesen Bericht verwenden, um zu ermitteln, wo Sie in Ihrer Amazon RDS-DB-Instance ein Schema erstellen müssen, das mit Ihrer Quelldatenbank kompatibel ist.
- Schemas mit AWS SCT konvertieren— Wenn Sie Schemaelemente haben, die nicht automatisch konvertiert werden können, haben Sie zwei Möglichkeiten: Aktualisieren Sie das Quellschema und konvertieren Sie es erneut, oder erstellen Sie entsprechende Schemaelemente in Ihrer Amazon RDS-DB-Ziel-Instance.
- Aktualisierung und Aktualisierung der konvertierten Schemas in AWS SCT— Sie können Ihr AWS SCT Projekt mit dem neuesten Schema aus Ihrer Quelldatenbank aktualisieren.
- Speichern und Anwenden konvertierter Schemas in AWS SCT— Wenn Sie bereit sind, AWS SCT wenden Sie das konvertierte Schema in Ihrem lokalen Projekt auf Ihre Amazon RDS-DB-Ziel-Instance an.

Anwenden von Migrationsregeln in AWS Schema Conversion Tool

Bevor Sie Ihr Schema mit konvertieren AWS SCT, können Sie Migrationsregeln einrichten. Migrationsregeln in AWS SCT können beispielsweise den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern. Nehmen wir beispielsweise an, dass Sie in Ihrem Quellschema eine Reihe von Tabellen mit dem Namen `test_TABLE_NAME` haben. Sie können eine Regel einrichten, die das Präfix `test_` im Zielschema in das Präfix `demo_` ändert.

> **Note**
>
> Sie können Migrationsregeln nur für verschiedene Quell- und Zieldatenbank-Engines erstellen.

Sie können Migrationsregeln erstellen, die die folgenden Aufgaben ausführen:

- Ein Präfix hinzufügen, entfernen oder ersetzen
- Ein Suffix hinzufügen, entfernen oder ersetzen
- Spaltensortierung ändern
- Ändern des Datentyps
- Ändern Sie die Länge von`char`, `varcharnvarchar`, und `string` Datentypen
- Verschieben von Objekten
- Umbenennen von Objekten

Sie können Migrationsregeln für die folgenden Objekte erstellen:

- Datenbank
- Schema
- Tabelle
- Spalte

Migrationsregeln erstellen

Sie können Migrationsregeln erstellen und die Regeln als Teil Ihres Projekts speichern. Gehen Sie bei geöffnetem Projekt wie folgt vor, um Migrationsregeln zu erstellen.

Um Migrationsregeln zu erstellen

1. Wählen Sie im Menü Ansicht die Option Kartenansicht aus.
2. Wählen Sie unter Serverzuordnungen ein Paar aus Quell- und Zielservern aus.
3. Wählen Sie Neue Migrationsregel aus. Das Dialogfeld „Transformationsregeln“ wird angezeigt.
4. Wählen Sie Add new rule aus. Der Liste der Regeln wird eine neue Zeile hinzugefügt.
5. Konfigurieren Sie Ihre Regel:

a. Geben Sie unter Name einen Namen für Ihre Regel ein.

b. Wählen Sie für For den Objekttyp aus, für den die Regel gelten soll.

c. Geben Sie für den Ort einen Filter ein, der auf Objekte angewendet werden soll, bevor Sie die Migrationsregel anwenden. Die WHERE-Klausel wird mithilfe einer vergleichbaren Klausel ausgewertet. Sie können den exakten Namen eingeben und ein Objekt auswählen oder Sie können ein Muster angeben und damit mehrere Objekte auswählen.

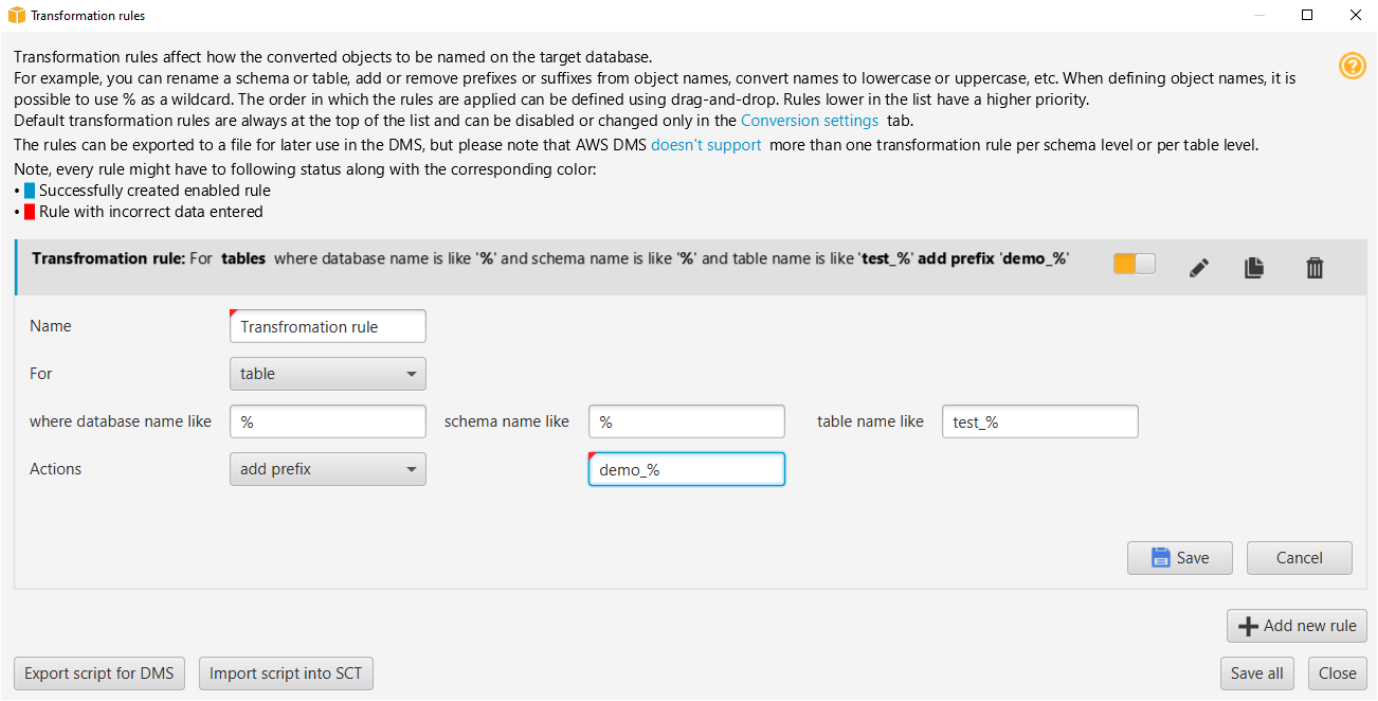
   Die verfügbaren Felder für die Klausel where sind je nach Objekttyp unterschiedlich. Wenn als Objekttyp "Schema" ausgewählt wurde, ist nur das Feld für den Schemanamen verfügbar.

d. Wählen Sie unter Aktionen den Typ der Migrationsregel aus, die Sie erstellen möchten.

e. Geben Sie je nach Regeltyp einen oder zwei zusätzliche Werte ein. Wenn Sie beispielsweise ein Objekt umbenennen möchten, geben Sie den neuen Namen des Objekts ein. Wenn Sie ein Präfix ersetzen möchten, geben Sie das alte und das neue Präfix ein.

   Für die Datentypen char, varchar, nvarchar und string können Sie die Länge des Datentyps mithilfe des Multiplikationsoperators ändern. Beispielsweise wandelt der `%*4` Wert den Datentyp in. `varchar(10)` `varchar(40)`

6. Nachdem Sie Ihre Migrationsregel konfiguriert haben, wählen Sie Speichern, um Ihre Regel zu speichern. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.



7. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen der Regeln fertig sind, klicken Sie auf Save All, um alle Änderungen zu speichern.
8. Wählen Sie „Schließen“, um das Dialogfeld „Transformationsregeln“ zu schließen.

Sie können das Umschaltsymbol verwenden, um eine Migrationsregel zu deaktivieren, ohne sie zu löschen. Sie können das Kopiersymbol verwenden, um eine bestehende Migrationsregel zu duplizieren. Sie können das Stiftsymbol verwenden, um eine bestehende Migrationsregel zu bearbeiten. Sie können das Löschsymbol verwenden, um eine bestehende Migrationsregel zu löschen. Um alle Änderungen zu speichern, die Sie an Ihren Migrationsregeln vornehmen, wählen Sie Alle speichern.

Migrationsregeln exportieren

Wenn Sie AWS DMS früher Ihre Daten von Ihrer Quelldatenbank in Ihre Zieldatenbank migrieren, können Sie Informationen zu Ihren Migrationsregeln für bereitstellen AWS DMS. Weitere Informationen zu Aufgaben finden Sie unter Arbeiten mit AWS Database Migration Service Replikationsaufgaben.

Um Migrationsregeln zu exportieren

1. Wählen Sie in der AWS Schema Conversion Tool im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie unter Migrationsregeln eine Migrationsregel aus und klicken Sie dann auf Migrationsregel ändern.
3. Wählen Sie Skript exportieren für AWS DMS.
4. Navigieren Sie zum gewünschten Speicherort des Skripts und klicken Sie auf Save. Ihre Migrationsregeln werden als JSON-Skript gespeichert, das von verwendet werden kann AWS DMS.

Schemas mit AWS SCT konvertieren

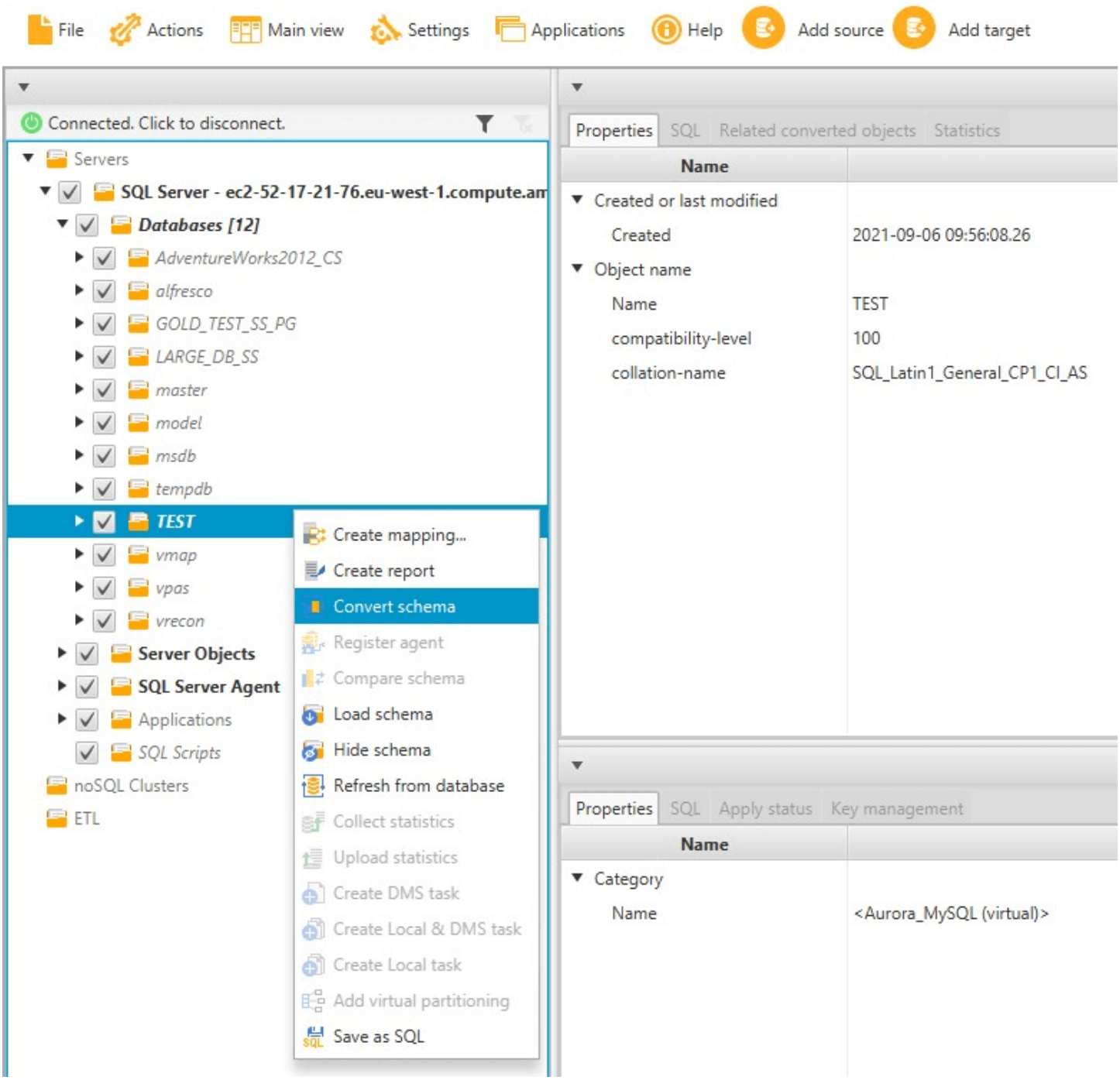
Nachdem Sie Ihr Projekt sowohl mit Ihrer Quelldatenbank als auch mit Ihrer Amazon RDS-DB-Ziel-Instance verbunden haben, zeigt Ihr AWS Schema Conversion Tool Projekt das Schema aus Ihrer Quelldatenbank im linken Bereich an. Das Schema wird in einer Strukturansicht dargestellt und jeder Knoten wird per Lazy Loading abgerufen. Wenn Sie einen Knoten in der Strukturansicht auswählen,

AWS SCT werden zu diesem Zeitpunkt die Schemainformationen aus Ihrer Quelldatenbank angefordert.

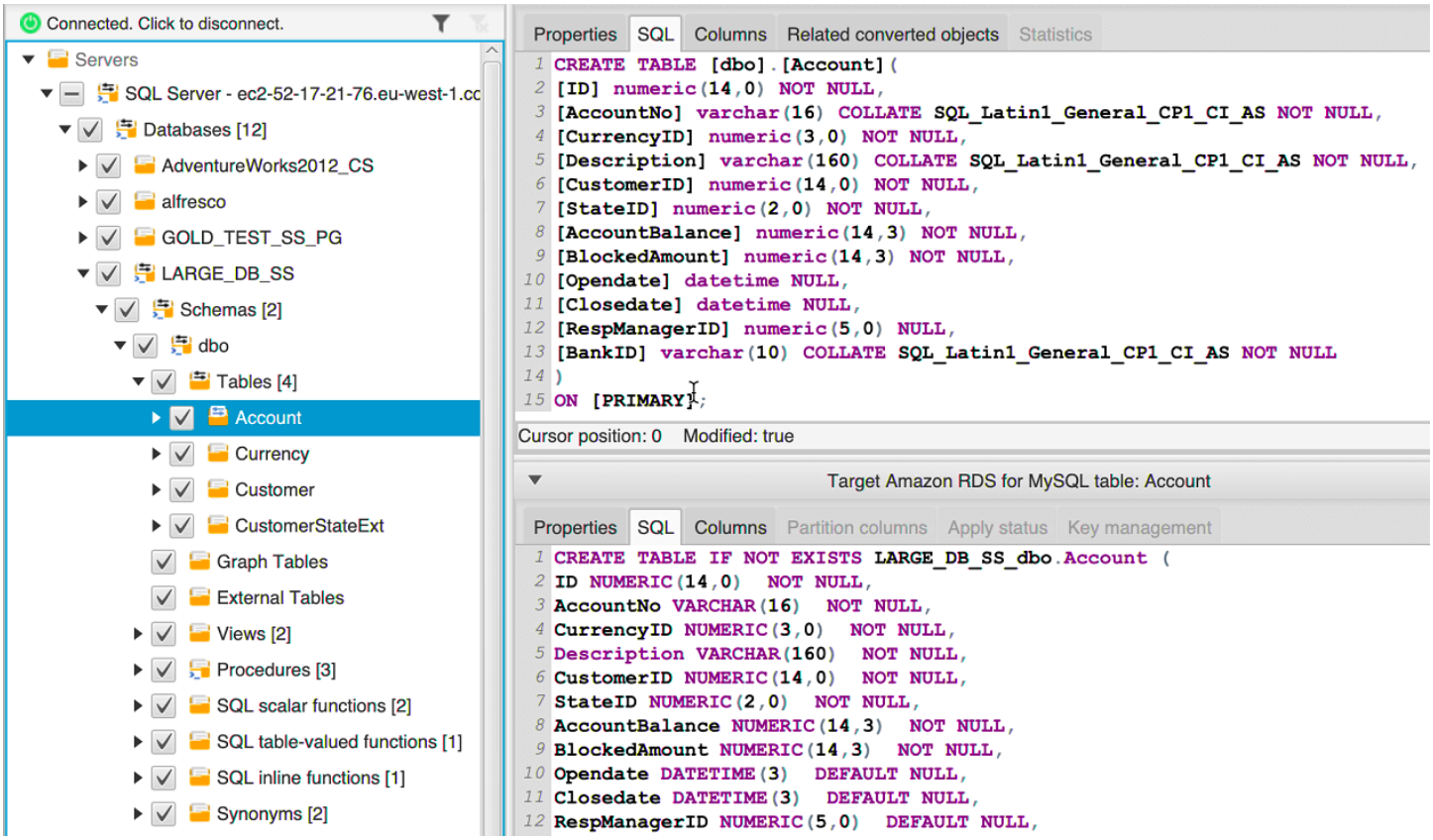
Sie können Schemaelemente in der Quelldatenbank auswählen und dann das Schema in ein entsprechendes Schema für die DB-Engine der DB-Ziel-Instance konvertieren. Sie können beliebige Schemaelemente aus der Quelldatenbank für die Konvertierung auswählen. Wenn das von Ihnen gewählte Schemaelement von einem übergeordneten Element abhängt, generiert es AWS SCT auch das Schema für das übergeordnete Element. Nehmen wir beispielsweise an, Sie wählen eine Tabelle für die Konvertierung aus. Falls ja, AWS SCT generiert das Schema für die Tabelle und die Datenbank, in der sich die Tabelle befindet.

Konvertieren des Schemas

Um ein Schema aus Ihrer Quelldatenbank zu konvertieren, aktivieren Sie das Kontrollkästchen für den Namen des zu konvertierenden Schemas. Wählen Sie als Nächstes dieses Schema im linken Bereich Ihres Projekts aus. AWS SCT hebt den Schemanamen blau hervor. Öffnen Sie das Kontextmenü (Rechtsklick) für das Schema und wählen Sie Schema konvertieren aus, wie im Folgenden dargestellt.



Nachdem Sie das Schema von der Quelldatenbank konvertiert haben, können Sie aus dem linken Projektbereich Schemaelemente auswählen und das konvertierte Schema im mittleren Projektbereich anzeigen. Im mittleren Bereich werden unten die Eigenschaften des konvertierten Schemas und der SQL-Befehl zum Erstellen desselben wie folgt angezeigt:



Nachdem das Schema konvertiert ist, können Sie das Projekt speichern. Die Schemainformationen von der Quelldatenbank werden zusammen mit dem Projekt gespeichert. Dank dieser Funktionalität können Sie offline ohne Verbindung zur Quelldatenbank arbeiten. AWS SCT stellt eine Verbindung zur Quelldatenbank her, um das Schema in Ihrem Projekt zu aktualisieren, wenn Sie Refresh from Database (Aktualisieren von Datenbank) für Ihre Quelldatenbank wählen. Weitere Informationen finden Sie unter Aktualisierung und Aktualisierung der konvertierten Schemas in AWS SCT.

Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, in dem die nicht automatisch konvertierbaren Elemente aufgeführt werden. Mit diesem Bewertungsbericht können Sie die Schemaelemente ermitteln und auflösen, die nicht automatisch konvertiert werden können. Weitere Informationen finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.

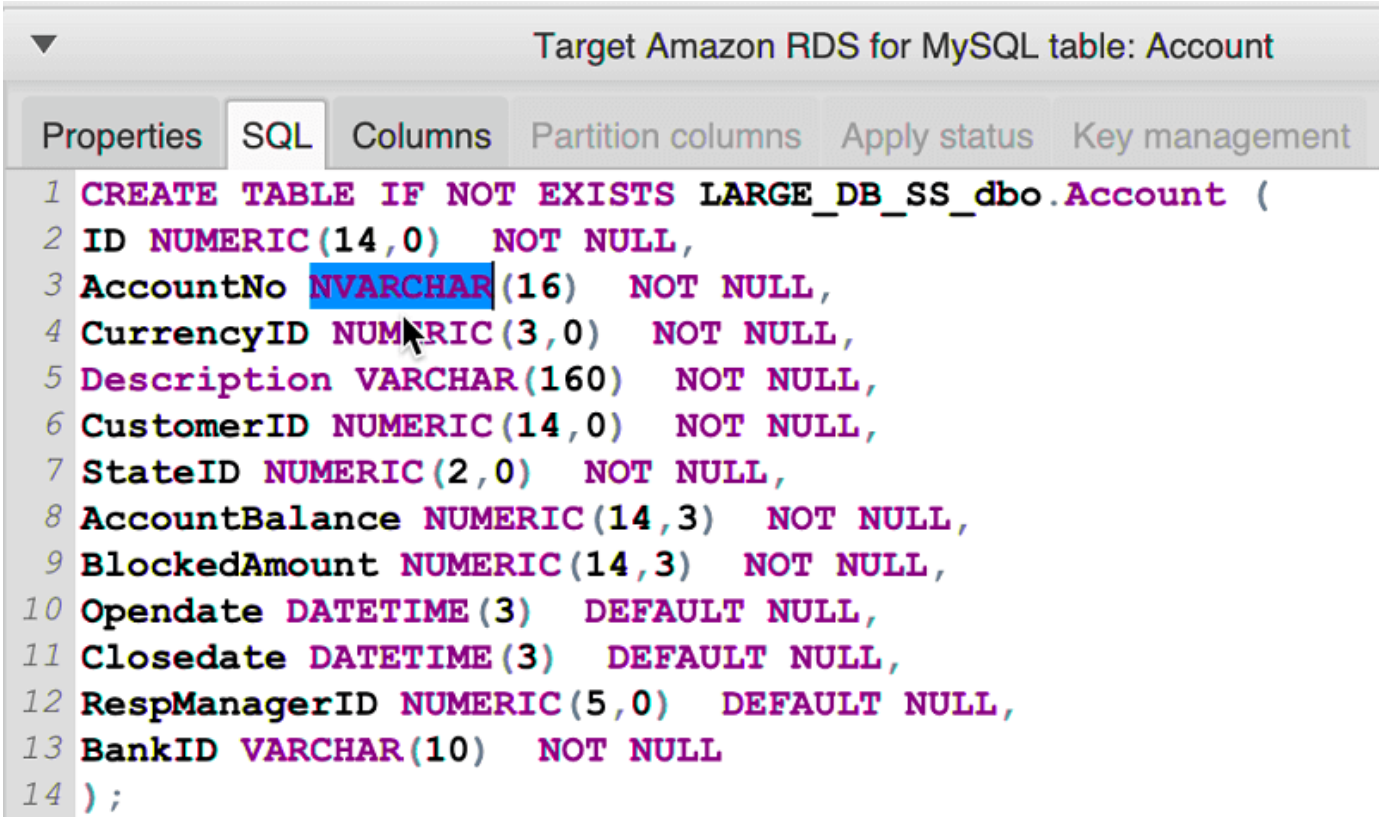
Wenn ein konvertiertes Schema AWS SCT generiert wird, wird es nicht sofort auf die Ziel-DB-Instance angewendet. Stattdessen wird das konvertierte Schema lokal gespeichert, bis Sie es für die DB-Ziel-Instance übernehmen. Weitere Informationen finden Sie unter Ihr konvertiertes Schema wird angewendet.

Konvertiertes Schema bearbeiten

Sie können ein konvertiertes Schema bearbeiten und die Änderungen als Teil des Projekts speichern.

So bearbeiten Sie ein konvertiertes Schema

1. Wählen Sie im linken Projektbereich, in dem das Schema von der Quelldatenbank angezeigt wird, das Schemaelement aus, für das Sie das konvertierte Schema ändern möchten.
2. Wählen Sie unten im mittleren Bereich, in dem das konvertierte Schema für das ausgewählte Element angezeigt wird, die Registerkarte SQL aus.
3. Nehmen Sie im Text, der unter der Registerkarte SQL angezeigt wird, die erforderlichen Änderungen am Schema vor. Das Schema wird automatisch für das Projekt gespeichert, wenn Sie es aktualisieren.



Die Änderungen, die Sie am konvertierten Schema vorgenommen haben, werden für das Projekt gespeichert, wenn Sie es aktualisieren. Wenn Sie ein Schemaelement aus der Quelldatenbank neu konvertieren, aber das zuvor konvertierte Schema für dieses Element bereits aktualisiert hatten, werden die erfolgten Aktualisierungen durch das auf Basis der Quelldatenbank neu konvertierte Schemaelement ersetzt.

Ein konvertiertes Schema löschen

Bis Sie das Schema auf Ihre Ziel-DB-Instance anwenden, wird das konvertierte Schema AWS SCT nur lokal in Ihrem Projekt gespeichert. Sie können das geplante Schema aus Ihrem Projekt löschen, indem Sie den Tree-View-Knoten für Ihre DB-Instance auswählen und dann Aus Datenbank aktualisieren wählen. Da kein Schema in Ihre Ziel-DB-Instance geschrieben wurde, werden bei der Aktualisierung aus der Datenbank die geplanten Schemaelemente in Ihrem AWS SCT Projekt entfernt, sodass sie mit denen in Ihrer Quell-DB-Instance übereinstimmen.

Manuelles Konvertieren von Schemas in AWS SCT

Der Bewertungsbericht enthält eine Liste von Elementen, die nicht automatisch in die Datenbank-Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Für jedes nicht konvertierbare Element wird auf der Registerkarte Action Items ein Aktionselement angegeben.

Sie können wie folgt auf die Aktionselemente im Bewertungsbericht reagieren:

- Sie ändern das Schema der Quelldatenbank.
- Sie ändern das Schema der Zieldatenbank.

Ändern Sie Ihr Quellschema

Bei einigen Elementen ist es möglicherweise einfacher, das Datenbankschema in Ihrer Quelldatenbank in ein Schema zu ändern, das automatisch konvertiert werden kann. Vergewissern Sie sich zunächst, dass die neuen Änderungen mit der Anwendungsarchitektur kompatibel sind, und aktualisieren Sie dann das Schema der Quelldatenbank. Aktualisieren Sie abschließend das Projekt mit den neuen Schemainformationen. Anschließend können Sie das aktualisierte Schema konvertieren und den Bewertungsbericht zur Datenbankmigration erneut erstellen. Für die im Quellschema geänderten Elemente werden keine Aktionselemente mehr angezeigt.

Der Vorteil dieses Verfahrens besteht darin, dass das aktualisierte Schema immer verfügbar ist, wenn Sie von der Quelldatenbank aktualisieren.

Ändern Sie Ihr Zielschema

Bei einigen Elementen ist es einfacher, das konvertierte Schema für die Zieldatenbank zu übernehmen und anschließend für die Elemente, die nicht automatisch konvertiert werden konnten,

entsprechende Schemaelemente manuell zur Zieldatenbank hinzuzufügen. Durch Übernehmen des jeweiligen Schemas können Sie automatisch konvertierbare Schemata in die DB-Ziel-Instance schreiben. Weitere Informationen finden Sie unter Speichern und Anwenden konvertierter Schemas in AWS SCT.

Das Schema, das in die DB-Ziel-Instance geschrieben wird, enthält nur automatisch konvertierbare Elemente. Nachdem Sie das Schema für die DB-Ziel-Instance übernommen haben, können Sie manuell ein (mit der Quelldatenbank) gleichwertiges Schema in der DB-Ziel-Instance erstellen. Die im Bewertungsbericht zur Datenbankmigration angegebenen Aktionselemente enthalten Vorschläge zum Erstellen eines gleichwertigen Schemas.

> **Warning**
>
> Wenn Sie ein Schema manuell in der DB-Ziel-Instance erstellen, sollten Sie eine Kopie davon speichern. Falls Sie das konvertierte Schema vom Projekt erneut in die DB-Ziel-Instance übernehmen, werden Ihre manuellen Änderungen überschrieben.

In einigen Fällen ist es nicht möglich, ein gleichwertiges Schema in der DB-Ziel-Instance zu erstellen. Möglicherweise muss das Design der Anwendung und der Datenbank in Teilen überarbeitet werden, damit Sie die von der DB-Engine bereitgestellte Funktionalität für Ihre DB-Ziel-Instance nutzen können. In anderen Fällen kann das Schema, das nicht automatisch konvertiert werden kann, ignoriert werden.

Aktualisierung und Aktualisierung der konvertierten Schemas in AWS SCT

Sie können sowohl das Quellschema als auch das Zielschema in Ihrem AWS Schema Conversion Tool Projekt aktualisieren.

- Quelle — Wenn Sie das Schema für Ihre Quelldatenbank aktualisieren, AWS SCT wird das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Quelldatenbank ersetzt. Mit dieser Funktion können Sie das Projekt aktualisieren, wenn das Schema der Quelldatenbank geändert wurde.
- Ziel — Wenn Sie das Schema für Ihre Ziel-Amazon-RDS-DB-Instance aktualisieren, wird das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Ziel-DB-Instance AWS SCT ersetzt. Wenn Sie kein Schema auf Ihre Ziel-DB-Instance angewendet haben, AWS SCT löscht

das konvertierte Schema aus Ihrem Projekt. Anschließend können Sie das Schema von der Quelldatenbank für eine neue DB-Ziel-Instance konvertieren.

Sie aktualisieren das Schema in Ihrem AWS SCT Projekt, indem Sie Aus Datenbank aktualisieren wählen.

> **Note**
>
> Wenn Sie Ihr Schema aktualisieren, werden Metadaten nur dann AWS SCT geladen, wenn sie benötigt werden. Um das gesamte Schema Ihrer Datenbank vollständig zu laden, öffnen Sie das Kontextmenü (Rechtsklick) für Ihr Schema und wählen Sie Schema laden. Sie können diese Option beispielsweise verwenden, um Metadaten für Ihre Datenbank auf einmal zu laden und dann offline zu arbeiten.

Speichern und Anwenden konvertierter Schemas in AWS SCT

Wenn das konvertierte Schema AWS Schema Conversion Tool generiert (wie unter gezeigt???), wendet es das konvertierte Schema nicht sofort auf die Ziel-DB-Instance an. Stattdessen wird das konvertierte Schema lokal im Projekt gespeichert, bis Sie es für die DB-Ziel-Instance übernehmen. Mit dieser Funktion können Sie mit Schemaelementen arbeiten, die sich nicht automatisch in die Zieldatenbank-Engine konvertieren lassen. Weitere Informationen zu Elementen, die nicht automatisch konvertierbar sind, finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.

Optional können Sie das konvertierte Schema mithilfe des Tools als SQL-Skript in einer Datei speichern, bevor Sie das Schema für die DB-Ziel-Instance übernehmen. Mit diesem Tool können Sie das konvertierte Schema auch direkt für die DB-Ziel-Instance übernehmen.

Speichern Sie Ihr konvertiertes Schema in einer Datei

Sie können das konvertierte Schema als SQL-Skripts in einer Textdatei speichern. Mit diesem Ansatz können Sie die generierten SQL-Skripts ändern, um Elemente AWS SCT zu adressieren, die das Tool nicht automatisch konvertieren kann. Anschließend führen Sie die aktualisierten Skripts auf der DB-Ziel-Instance aus und übernehmen das konvertierte Schema in die Zieldatenbank.

So speichern Sie Ihr konvertiertes Schema als SQL-Skripts

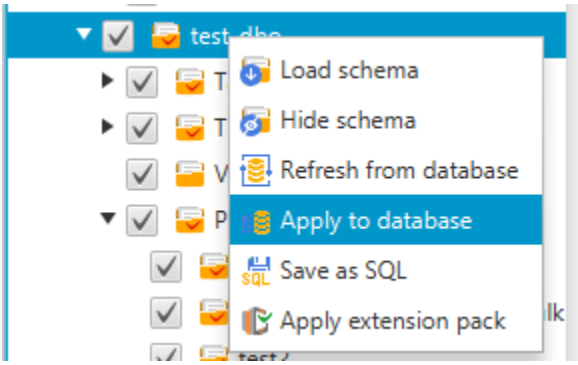
1. Wählen Sie Ihr Schema und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie Als SQL speichern.
3. Geben Sie den Namen der Datei ein und wählen Sie Speichern.
4. Speichern Sie Ihr konvertiertes Schema mit einer der folgenden Optionen:
   - Einzelne Datei
   - Eine Datei pro Phase
   - Eine Datei pro Kontoauszug

Um das Format des SQL-Skripts auszuwählen

1. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen aus.
2. Wählen Sie „Skripts speichern“.
3. Wählen Sie für Vendor die Datenbankplattform aus.
4. Wählen Sie unter SQL-Skripten speichern unter aus, wie Sie Ihr Datenbankschemaskript speichern möchten.
5. Wählen Sie OK, um die Einstellungen zu speichern.

Ihr konvertiertes Schema wird angewendet

Wenn Sie bereit sind, Ihr konvertiertes Schema auf Ihre Amazon RDS-DB-Zielinstanz anzuwenden, wählen Sie das Schemaelement im rechten Bereich Ihres Projekts aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schemaelement und wählen Sie Apply to database aus, wie hier gezeigt.



Das Erweiterungspaket-Schema

Wenn Sie zum ersten Mal ein konvertiertes Schema für die DB-Ziel-Instance übernehmen, fügt AWS SCT ein weiteres Schema zu Ihrer DB-Ziel-Instance hinzu. Über dieses Schema werden Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Ziel-DB-Instance benötigt werden. Das Schema wird als Erweiterungspaketschema bezeichnet.

Dieses Erweiterungspaketschema sollten Sie nicht ändern, andernfalls könnten im konvertierten Schema, das auf die DB-Ziel-Instance geschrieben wird, unerwünschte Ergebnisse auftreten. Wenn Ihr Schema vollständig auf Ihre Ziel-DB-Instance migriert ist und Sie es nicht mehr benötigen AWS SCT, können Sie das Erweiterungspaket-Schema löschen.

Das Erweiterungspaketschema wird wie folgt entsprechend Ihrer Quelldatenbank benannt:

- IBM Db2 LUW: `aws_db2_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- SAP-FALL: `aws_sapase_ext`

Weitere Informationen finden Sie unter Verwenden der AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket .

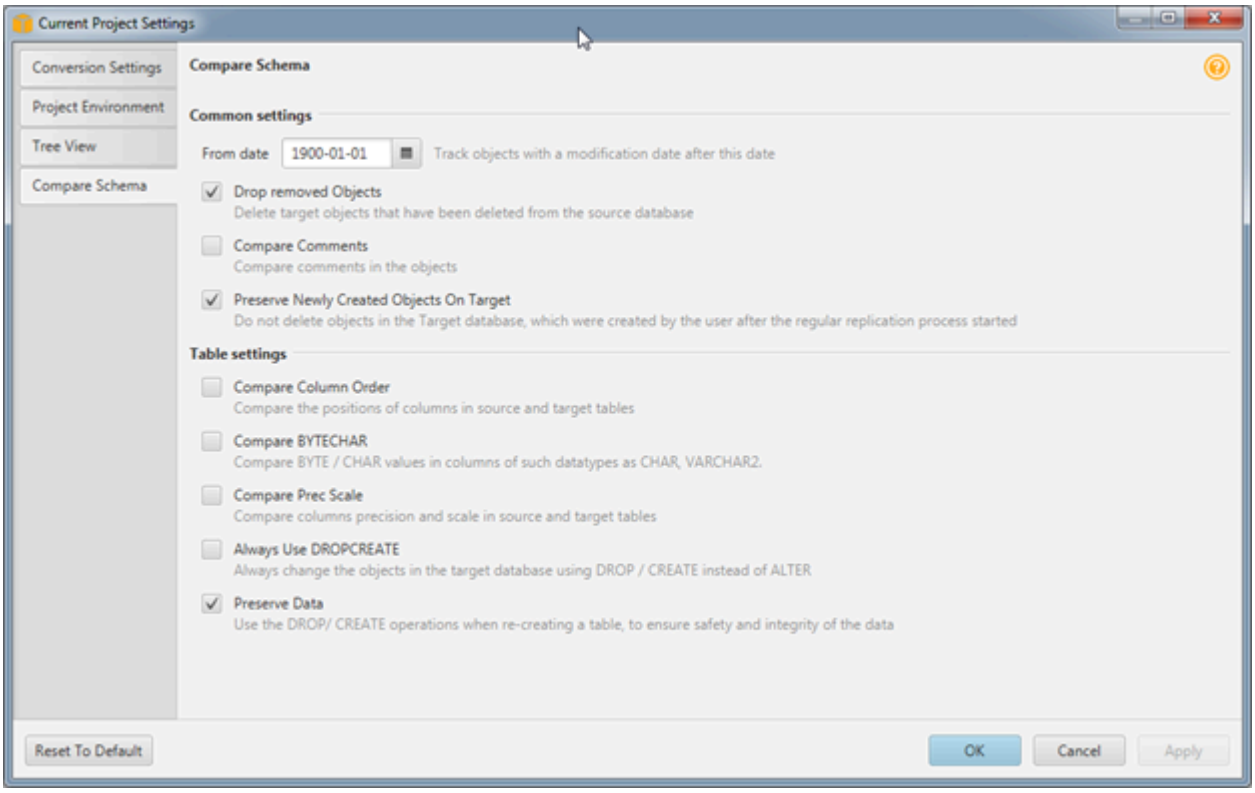
Schemas vergleichen in AWS Schema Conversion Tool

Wenn Sie nach der Migration Änderungen an Ihrem Quell- oder Zielschema vorgenommen haben, können Sie die beiden Datenbankschemas mithilfe von vergleichen. AWS SCT Sie können Schemata der gleichen Version vergleichen oder eine Version mit einer früheren Version des Quellschemas.

Die folgenden Schemavergleiche werden unterstützt:

- Oracle mit Oracle, Versionen 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0 und 10
- SQL Server zu SQL Server, Versionen 2016, 2014, 2012, 2008, 2008 RD2
- PostgreSQL to PostgreSQL und Aurora PostgreSQL-kompatible Edition, Versionen 9.6, 9.5.9, 9.5.4
- MySQL mit MySQL, Versionen 5.6.36, 5.7.17 und 5.5

Sie geben Einstellungen für den Schemavergleich auf der Registerkarte Compare Schema (Schema vergleichen) der Seite Project Settings (Projekteinstellungen) an.

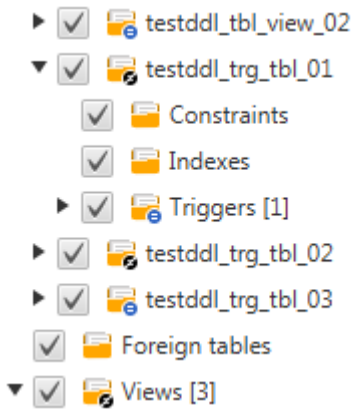


Um Schemas zu vergleichen, wählen Sie die Schemas aus und geben die Objekte an, die sich zwischen den beiden Schemas unterscheiden, und die Objekte, bei AWS SCT denen dies nicht der Fall ist.

So vergleichen Sie zwei Schemata

1. Öffnen Sie ein vorhandenes AWS SCT -Projekt oder erstellen Sie ein Projekt und stellen Sie eine Verbindung zu den Quell- und Zielendpunkten her.
2. Wählen Sie das zu vergleichende Schema aus.
3. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Schema vergleichen.

AWS SCT kennzeichnet Objekte, die sich zwischen den beiden Schemas unterscheiden, indem dem Objektsymbol ein schwarzer Kreis hinzugefügt wird.



Sie können die Ergebnisse des Schemavergleichs auf ein einzelnes Objekt, auf eine einzelne Kategorie von Objekten oder auf das gesamte Schema anwenden. Aktivieren Sie das Kontrollkästchen neben der Kategorie, dem Objekt oder dem Schema, auf die bzw. das die Ergebnisse angewendet werden sollen.

Anzeige verwandter transformierter Objekte in AWS Schema Conversion Tool

Nach einer Schemakonvertierung wurden in einigen Fällen AWS SCT möglicherweise mehrere Objekte für ein Schemaobjekt in der Quelldatenbank erstellt. Wenn Sie beispielsweise eine Konvertierung von Oracle nach PostgreSQL durchführen, AWS SCT nimmt jeder Oracle-Trigger und wandelt ihn in einen Trigger und eine Triggerfunktion auf dem PostgreSQL-Ziel um. Wenn eine Oracle-Paketfunktion oder -prozedur nach PostgreSQL AWS SCT konvertiert wird, erstellt sie außerdem eine äquivalente Funktion und eine INIT-Funktion, die als Init-Block ausgeführt werden sollten, bevor die Prozedur oder Funktion ausgeführt werden kann.

Mit dem folgenden Verfahren können Sie alle verwandten Objekte anzeigen, die während der Schemakonvertierung erstellt wurden.

So zeigen Sie verwandte Objekte an, die während der Schemakonvertierung erstellt wurden

1. Wählen Sie nach der Schemakonvertierung das konvertierte Objekt in der Zielstrukturansicht aus.
2. Wählen Sie die Registerkarte Related Converted Objects (Verwandte konvertierte Objekte) aus.
3. Sehen Sie sich die Liste der verwandten Zielobjekte an.

Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT

Die AWS Schema Conversion Tool (AWS SCT) automatisiert einen Großteil des Prozesses der Konvertierung Ihres Data Warehouse-Schemas in ein Amazon RDS- oder Aurora-Datenbankschema. Da die Quell- und Zieldatenbank-Engines viele verschiedene Funktionen und Fähigkeiten haben können, wird AWS SCT versucht, wo immer möglich, ein äquivalentes Schema in Ihrer Zieldatenbank zu erstellen. Wenn keine direkte Konvertierung möglich ist, AWS SCT wird ein Bewertungsbericht mit einer Liste möglicher Maßnahmen bereitgestellt, die Sie ergreifen können. Mithilfe AWS SCT können Sie Schlüssel verwalten, Datentypen und Objekte zuordnen und manuelle Konvertierungen vornehmen.

AWS SCT kann die folgenden Data Warehouse-Schemas konvertieren.

- Amazon Redshift
- Azure Synapse Analytics (Version 10)
- BigQuery
- Greenplum-Datenbank (Version 4.3)
- Microsoft SQL Server (Version 2008 und höher)
- Netezza (Version 7.0.3 und höher)
- Oracle (Version 10.2 und höher)
- Schneeflocke (Version 3)
- Teradata (Version 13 und höher)
- Vertica (Version 7.2 und höher)

Hinweise zur Konvertierung eines OLTP-Datenbankschemas (Online Transaction Processing) finden Sie unter. Datenbankschemas konvertieren in AWS Schema Conversion Tool

Gehen Sie wie folgt vor, um ein Data Warehouse-Schema zu konvertieren:

1. Geben Sie die Optimierungsstrategie und die Regeln sowie die Migrationsregeln an, die Sie verwenden AWS SCT möchten. Sie können Regeln festlegen, um den Datentyp von Spalten zu ändern, Objekte von einem Schema in ein anderes zu verschieben und Objektnamen zu bearbeiten.

Sie können Optimierungs- und Migrationsregeln in den Einstellungen angeben. Weitere Informationen zu Optimierungsstrategien finden Sie unterAuswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT. Weitere Informationen zu Migrationsregeln finden Sie unter Migrationsregeln erstellen in AWS SCT

2. Geben Sie Statistiken zu Ihrem Quell-Data Warehouse an, sodass AWS SCT optimieren kann, wie Ihr Data Warehouse umgewandelt wird. Sie können Statistiken direkt aus der Datenbank erfassen oder eine vorhandene Statistikdatei hochladen. Weitere Informationen über die Bereitstellung von Data-Warehouse-Statistiken finden Sie unter Sammeln oder Hochladen von Statistiken für AWS SCT.
3. Erstellen Sie einen Bewertungsbericht zur Datenbankmigration, in dem die Schemaelemente aufgeführt werden, die sich nicht automatisch umwandeln lassen. Mithilfe dieses Berichts können Sie ermitteln, wo Sie ein Schema in der Zieldatenbank manuell erstellen müssen, das kompatibel mit der Quelldatenbank ist. Weitere Informationen zum Bewertungsbericht finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.
4. Konvertieren Sie das Schema. AWS SCT erstellt eine lokale Version des konvertierten Schemas, damit Sie es prüfen können. Das Schema wird erst für die Zieldatenbank übernommen, wenn Sie mit der Überprüfung fertig sind. Weitere Informationen zur Umwandlung finden Sie unter Konvertieren Sie Ihr Schema mit AWS SCT.
5. Nachdem Sie das Schema konvertiert haben, können Sie Ihre Schlüssel verwalten und bearbeiten. Die Schlüsselverwaltung spielt eine zentrale Rolle bei der Data Warehouse-Konvertierung. Weitere Informationen zum Verwaltung von Schlüsseln finden Sie unter Schlüssel verwalten und anpassen in AWS SCT.
6. Falls Schemaelemente vorhanden sind, die nicht automatisch konvertiert werden können, haben Sie zwei Möglichkeiten: Entweder aktualisieren Sie das Quellschema und führen die Konvertierung erneut aus oder Sie erstellen entsprechende Schemaelemente in der Zieldatenbank. Weitere Informationen zum manuellen Umwandeln von Schemaelementen finden Sie unter Handhabung manueller Konvertierungen in AWS SCT. Weitere Informationen zur Aktualisierung Ihres Quellschemas finden Sie unter Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT.
7. Wenn Sie bereit sind, können Sie das umgewandelte Schema auf Ihre Zieldatenbank anwenden. Weitere Informationen zum Speichern und Anwenden des umgewandelten Schemas finden Sie unter Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT.

Berechtigungen für Amazon Redshift als Ziel

Die für Amazon Redshift als Ziel erforderlichen Berechtigungen sind im Folgenden aufgeführt:

- CREATE ON DATABASE — ermöglicht das Erstellen neuer Schemas in der Datenbank.
- CREATE ON SCHEMA — ermöglicht das Erstellen von Objekten im Datenbankschema.
- GRANT USAGE ON LANGUAGE — ermöglicht das Erstellen neuer Funktionen und Prozeduren in der Datenbank.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog — stellt dem Benutzer Systeminformationen über den Amazon Redshift Redshift-Cluster zur Verfügung.
- GRANT SELECT ON pg_class_info — liefert dem Benutzer Informationen über den Verteilungsstil von Tabellen.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu erteilen.

```
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;
```

Ersetzen Sie es im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie es dann *db_name* durch den Namen Ihrer Amazon Redshift Redshift-Zieldatenbank. Als Nächstes *schema_name* ersetzen Sie es durch den Namen Ihres Amazon Redshift Redshift-Schemas. Wiederholen Sie den `GRANT CREATE ON SCHEMA`-Vorgang für jedes Zielschema, auf das Sie den konvertierten Code anwenden oder zu dem Sie Daten migrieren möchten. Schließlich ersetzen Sie es *your_password* durch ein sicheres Passwort.

Sie können ein Erweiterungspaket auf Ihre Amazon-Redshift-Zieldatenbank anwenden. Ein Erweiterungspaket ist ein Add-On-Modul, das die Funktionen in der Quelldatenbank emuliert, die erforderlich sind, wenn Objekte zu Amazon Redshift konvertiert werden. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Für diesen Vorgang ist die Erlaubnis AWS SCT erforderlich, in Ihrem Namen auf den Amazon S3 S3-Bucket zuzugreifen. Um diese Berechtigung zu erteilen, erstellen Sie einen AWS Identity and Access Management (IAM-) Benutzer mit der folgenden Richtlinie.

```
{
   "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":""
      }
   ]
}
```

Auswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT

Um die AWS Schema Conversion Tool Konvertierung Ihres Data Warehouse-Schemas zu optimieren, können Sie die Strategien und Regeln auswählen, die das Tool verwenden soll. Nachdem Sie Ihr Schema konvertiert und die empfohlenen Schlüssel überprüft haben, können Sie Ihre Regeln anpassen oder die Strategie ändern, um die gewünschten Ergebnisse zu erzielen.

So legen Sie Optimierungsstrategien und -regeln fest

1. Wählen Sie Settings (Einstellungen) und anschließend Project Settings (Projekteinstellungen) aus. Das Dialogfeld Current project settings wird geöffnet.
2. Wählen Sie im linken Bereich Optimization Strategies aus. Die Optimierungsstrategien werden im rechten Bereich mit ihren Standardeinstellungen angezeigt.
3. Wählen Sie für Strategy Sector die gewünschte Optimierungsstrategie aus. Sie können aus den folgenden Optionen auswählen:

   - Metadaten verwenden, statistische Informationen ignorieren — Bei dieser Strategie werden nur Informationen aus den Metadaten für Optimierungsentscheidungen verwendet. Beispiel: Gibt es mehr als einen Index für eine Quelltabelle, wird die Sortierreihenfolge der Quelldatenbank verwendet und der erste Index wird zum Verteilungsschlüssel.

   - Metadaten ignorieren, statistische Informationen verwenden — Bei dieser Strategie werden Optimierungsentscheidungen ausschließlich anhand statistischer Informationen getroffen. Diese Strategie gilt nur für Tabellen und Spalten, für die Statistiken bereitgestellt werden. Weitere Informationen finden Sie unter Sammeln oder Hochladen von Statistiken für AWS SCT.

   - Metadaten verwenden und statistische Informationen verwenden — Bei dieser Strategie werden sowohl Metadaten als auch Statistiken für Optimierungsentscheidungen verwendet.

4. Nachdem Sie die Optimierungsstrategie festgelegt haben, können Sie die zu verwendenden Regeln auswählen. Sie können aus den folgenden Optionen auswählen:

   - Choose Distribution Key and Sort Keys using metadata
   - Choose fact table and appropriate dimension for collation
   - Analyze cardinality of indexes' columns
   - Suchen Sie in der Abfrage-Log-Tabelle nach den am häufigsten verwendeten Tabellen und Spalten

   Für jede Regel können Sie eine Gewichtung für den Sortierschlüssel und eine Gewichtung für den Verteilungsschlüssel eingeben. AWS SCT nutzt diese Gewichtungen beim Konvertieren

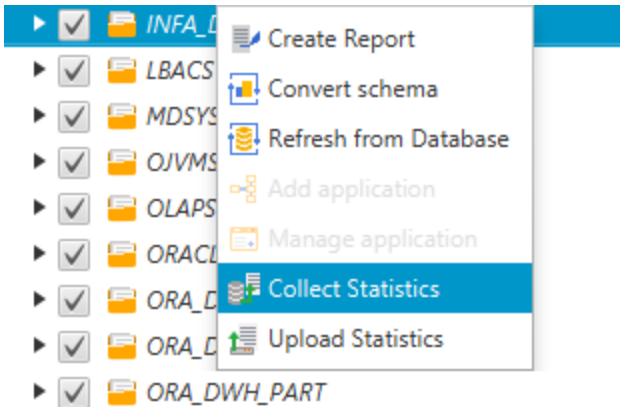
Ihres Schemas. Wenn Sie später die empfohlenen Schlüssel überprüfen, können Sie, wenn Sie mit den Ergebnissen nicht zufrieden sind, hierher zurückkehren, um Ihre Einstellungen zu ändern. Weitere Informationen finden Sie unter Schlüssel verwalten und anpassen in AWS SCT.

Sammeln oder Hochladen von Statistiken für AWS SCT

Um die AWS Schema Conversion Tool Konvertierung Ihres Data Warehouse-Schemas zu optimieren, können Sie Statistiken aus Ihrer Quelldatenbank bereitstellen, die das Tool verwenden kann. Sie können Statistiken direkt aus der Datenbank erfassen oder eine vorhandene Statistikdatei hochladen.

So stellen Sie Statistiken bereit und überprüfen diese

1. Öffnen Sie Ihr Projekt, und stellen Sie eine Verbindung zu Ihrer Quell-Datenbank her.
2. Wählen Sie ein Schemaobjekt aus dem linken Bereich Ihres Projekts aus und öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt. Wählen Sie, wie im Folgenden gezeigt, entweder Collect Statistics oder Upload Statistics aus.



3. Wählen ein Schemaobjekt aus dem linken Bereich Ihres Projekts und anschließend die Registerkarte Statistics aus. Sie können die statistischen Daten für das Objekt überprüfen.

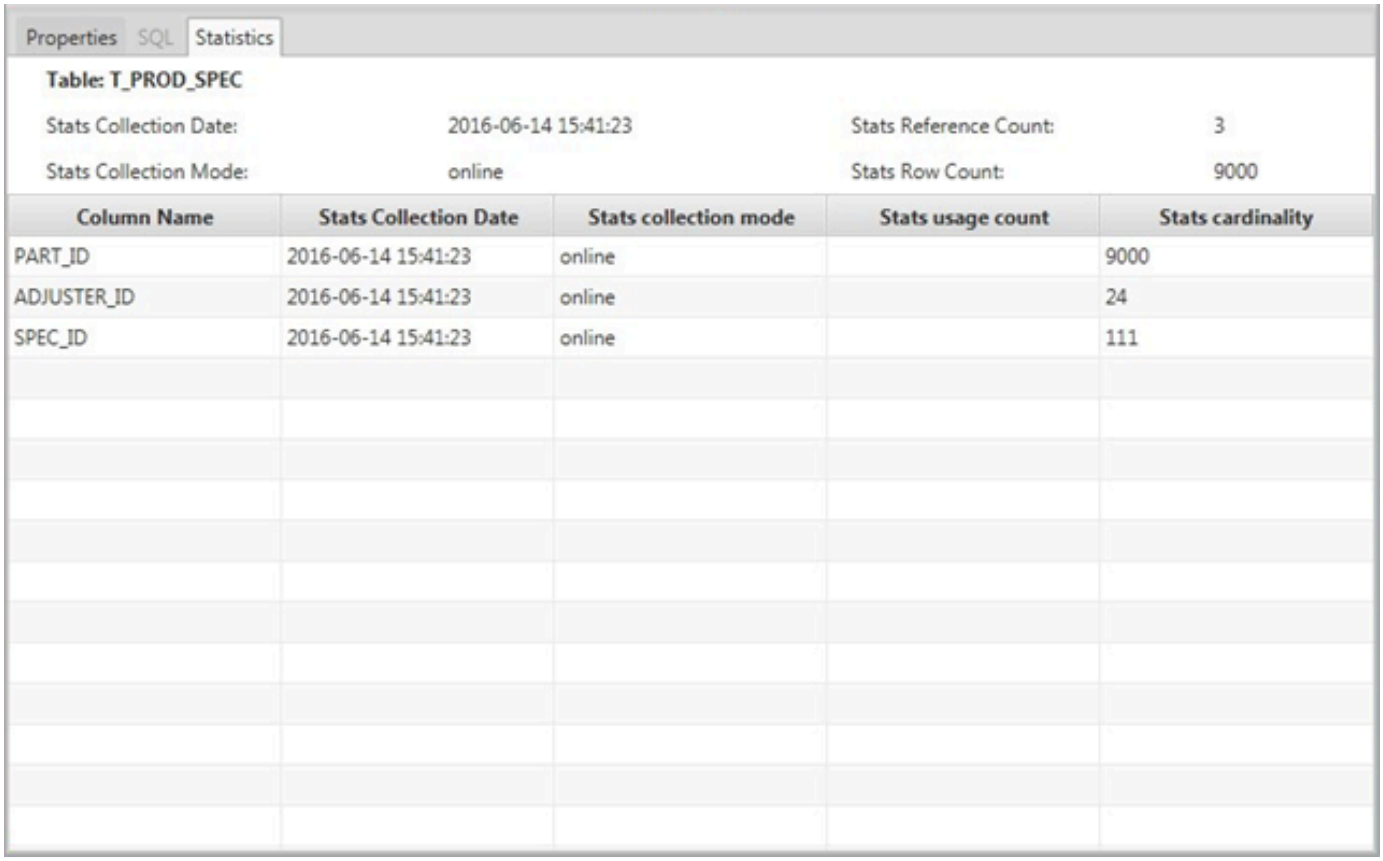

Properties SQL Statistics

Table: T_PROD_SPEC

Stats Collection Date: 2016-06-14 15:41:23 Stats Reference Count: 3

Stats Collection Mode: online Stats Row Count: 9000

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

Wenn Sie später die empfohlenen Schlüssel überprüfen, können Sie, wenn Sie mit den Ergebnissen nicht zufrieden sind, zusätzliche Statistiken erfassen und diesen Prozess wiederholen. Weitere Informationen finden Sie unter Schlüssel verwalten und anpassen in AWS SCT.

Migrationsregeln erstellen in AWS SCT

Bevor Sie Ihr Schema mit konvertieren AWS SCT, können Sie Migrationsregeln einrichten. Migrationsregeln können beispielsweise den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern. Nehmen wir beispielsweise an, dass Sie in Ihrem Quellschema eine Reihe von Tabellen mit dem Namen `test_TABLE_NAME` haben. Sie können eine Regel einrichten, die das Präfix `test_` im Zielschema in das Präfix `demo_` ändert.

> **Note**
>
> Sie können Migrationsregeln nur für verschiedene Quell- und Zieldatenbank-Engines erstellen.

Sie können Migrationsregeln erstellen, die die folgenden Aufgaben ausführen:

- Ein Präfix hinzufügen, entfernen oder ersetzen
- Ein Suffix hinzufügen, entfernen oder ersetzen
- Spaltensortierung ändern
- Ändern des Datentyps
- Ändern Sie die Länge von`char`, `varchar`nvarchar, und `string` Datentypen
- Verschieben von Objekten
- Umbenennen von Objekten

Sie können Migrationsregeln für die folgenden Objekte erstellen:

- Datenbank
- Schema
- Tabelle
- Spalte

Migrationsregeln erstellen

Sie können Migrationsregeln erstellen und die Regeln als Teil Ihres Projekts speichern. Gehen Sie bei geöffnetem Projekt wie folgt vor, um Migrationsregeln zu erstellen.

Um Migrationsregeln zu erstellen

1. Wählen Sie im Menü Ansicht die Option Kartenansicht aus.
2. Wählen Sie unter Serverzuordnungen ein Paar aus Quell- und Zielservern aus.
3. Wählen Sie Neue Migrationsregel aus. Das Dialogfeld „Transformationsregeln“ wird angezeigt.
4. Wählen Sie Add new rule aus. Der Liste der Regeln wird eine neue Zeile hinzugefügt.

5. Konfigurieren Sie Ihre Regel:

   a. Geben Sie unter Name einen Namen für Ihre Regel ein.

   b. Wählen Sie für For den Objekttyp aus, für den die Regel gelten soll.

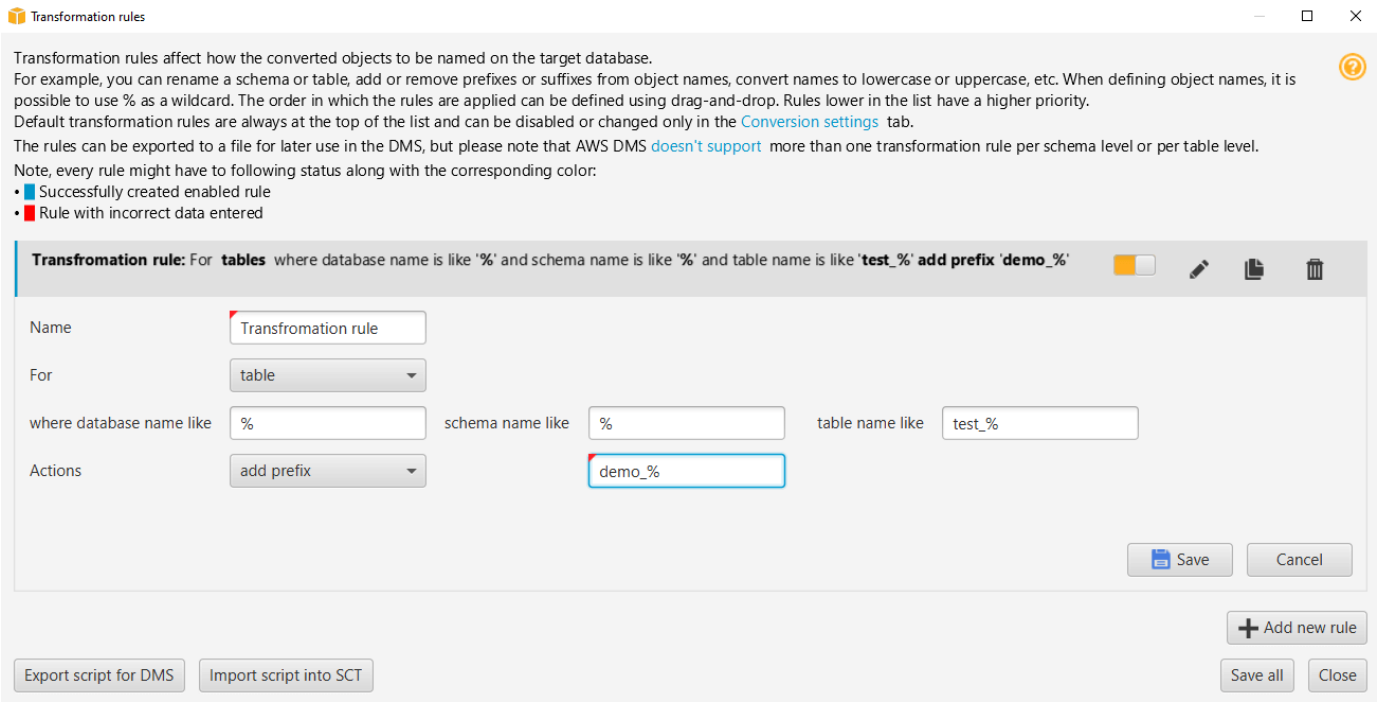
   c. Geben Sie für den Ort einen Filter ein, der auf Objekte angewendet werden soll, bevor Sie die Migrationsregel anwenden. Die WHERE-Klausel wird mithilfe einer vergleichbaren Klausel ausgewertet. Sie können den exakten Namen eingeben und ein Objekt auswählen oder Sie können ein Muster angeben und damit mehrere Objekte auswählen.

      Die verfügbaren Felder für die Klausel where sind je nach Objekttyp unterschiedlich. Wenn als Objekttyp "Schema" ausgewählt wurde, ist nur das Feld für den Schemanamen verfügbar.

   d. Wählen Sie unter Aktionen den Typ der Migrationsregel aus, die Sie erstellen möchten.

   e. Geben Sie je nach Regeltyp einen oder zwei zusätzliche Werte ein. Wenn Sie beispielsweise ein Objekt umbenennen möchten, geben Sie den neuen Namen des Objekts ein. Wenn Sie ein Präfix ersetzen möchten, geben Sie das alte und das neue Präfix ein.

6. Nachdem Sie Ihre Migrationsregel konfiguriert haben, wählen Sie Speichern, um Ihre Regel zu speichern. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.



7. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen der Regeln fertig sind, klicken Sie auf Save All, um alle Änderungen zu speichern.

8. Wählen Sie „Schließen“, um das Dialogfeld „Transformationsregeln“ zu schließen.

Sie können das Umschaltsymbol verwenden, um eine Migrationsregel zu deaktivieren, ohne sie zu löschen. Sie können das Kopiersymbol verwenden, um eine bestehende Migrationsregel zu duplizieren. Sie können das Stiftsymbol verwenden, um eine bestehende Migrationsregel zu bearbeiten. Sie können das Löschsymbol verwenden, um eine bestehende Migrationsregel zu löschen. Um alle Änderungen zu speichern, die Sie an Ihren Migrationsregeln vornehmen, wählen Sie Alle speichern.

Migrationsregeln exportieren

Wenn Sie AWS Database Migration Service (AWS DMS) verwenden, um Ihre Daten von Ihrer Quelldatenbank in Ihre Zieldatenbank zu migrieren, können Sie Informationen zu Ihren Migrationsregeln für bereitstellen AWS DMS. Weitere Informationen zu Aufgaben finden Sie unter Arbeiten mit AWS Database Migration Service Replikationsaufgaben.

Um Migrationsregeln zu exportieren

1. Wählen Sie in der AWS Schema Conversion Tool im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie unter Migrationsregeln eine Migrationsregel aus und klicken Sie dann auf Migrationsregel ändern.
3. Wählen Sie Skript exportieren für AWS DMS.
4. Navigieren Sie zum gewünschten Speicherort des Skripts und klicken Sie auf Save. Ihre Migrationsregeln werden als JSON-Skript gespeichert, das von verwendet werden kann AWS DMS.

Konvertieren Sie Ihr Schema mit AWS SCT

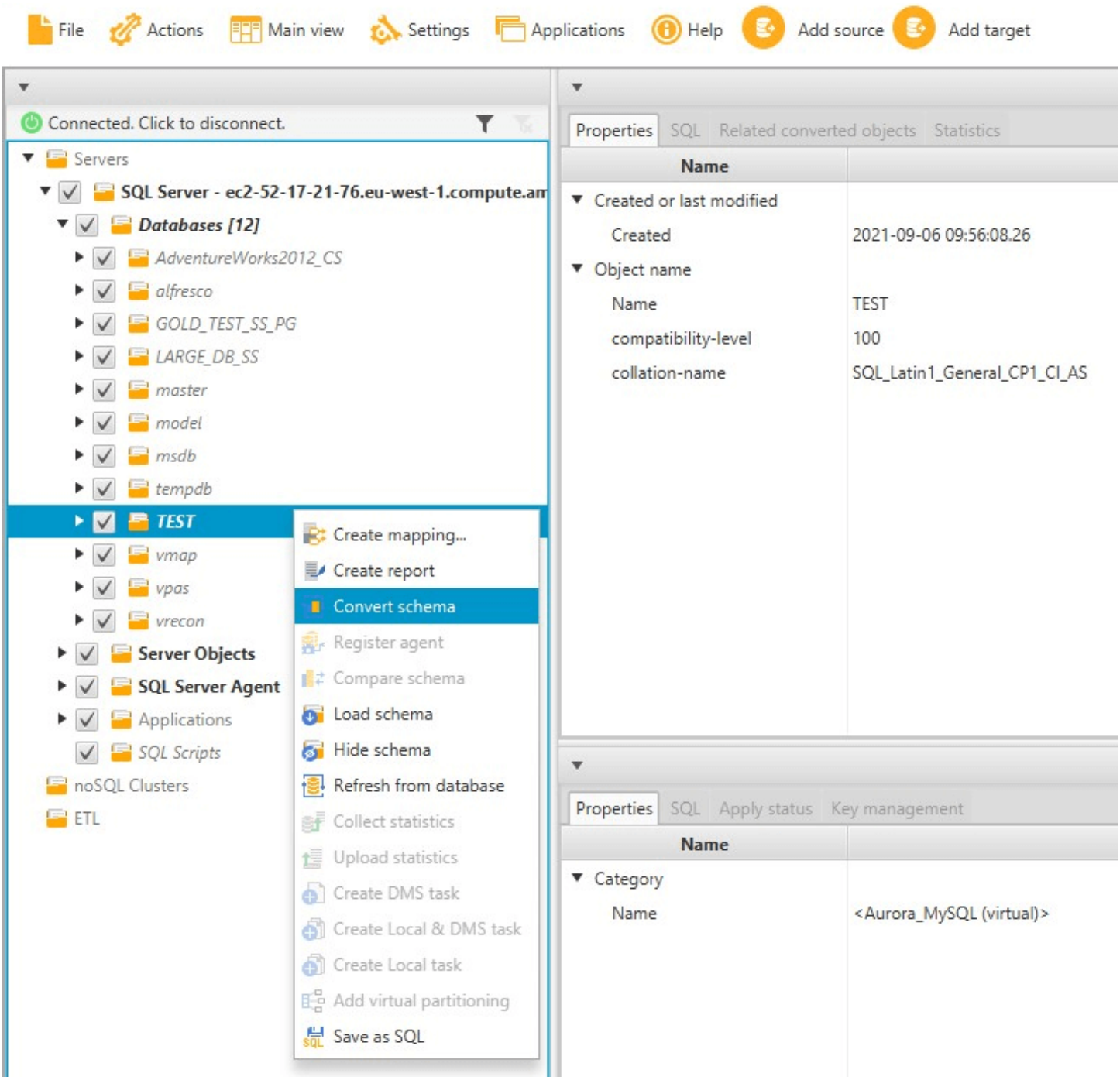
Nachdem Sie Ihr Projekt sowohl mit Ihrer Quelldatenbank als auch mit Ihrer Zieldatenbank verbunden haben, zeigt Ihr AWS Schema Conversion Tool Projekt das Schema aus Ihrer Quelldatenbank im linken Bereich an. Das Schema wird in einer Strukturansicht dargestellt und jeder Knoten wird per Lazy Loading abgerufen. Wenn Sie einen Knoten in der Strukturansicht auswählen, fordert AWS SCT die Schemainformationen zu diesem Zeitpunkt aus der Quelldatenbank an.

Sie können Schemaelemente in der Quelldatenbank auswählen und dann das Schema in ein entsprechendes Schema für die Datenbank-Engine der Zieldatenbank konvertieren. Sie können beliebige Schemaelemente aus der Quelldatenbank für die Konvertierung auswählen. Wenn das von Ihnen gewählte Schemaelement von einem übergeordneten Element abhängt, generiert es AWS SCT

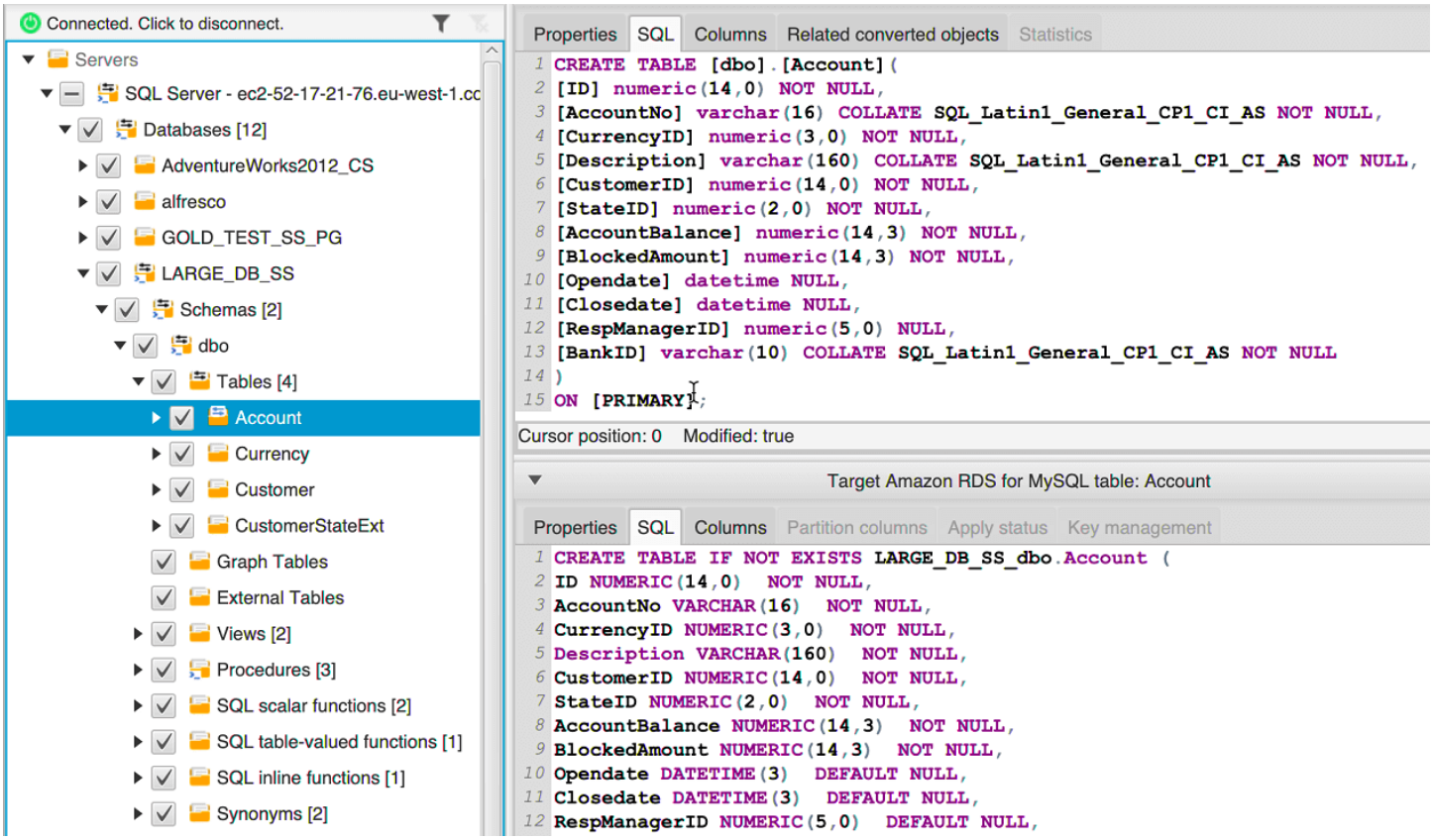
auch das Schema für das übergeordnete Element. Wenn Sie beispielsweise eine Spalte aus einer Tabelle auswählen, die konvertiert werden soll, werden das Schema für die Spalte, die Tabelle, in der sich die Spalte befindet, und die Datenbank, in der sich die Tabelle befindet, AWS SCT generiert.

Konvertieren des Schemas

Um ein Schema aus Ihrer Quelldatenbank zu konvertieren, aktivieren Sie das Kontrollkästchen für den Namen des zu konvertierenden Schemas. Wählen Sie als Nächstes dieses Schema im linken Bereich Ihres Projekts aus. AWS SCT hebt den Schemanamen blau hervor. Öffnen Sie das Kontextmenü (Rechtsklick) für das Schema und wählen Sie Schema konvertieren aus, wie im Folgenden dargestellt.



Nachdem Sie das Schema von der Quelldatenbank konvertiert haben, können Sie aus dem linken Projektbereich Schemaelemente auswählen und das konvertierte Schema im mittleren Projektbereich anzeigen. Im mittleren Bereich werden unten die Eigenschaften des konvertierten Schemas und der SQL-Befehl zum Erstellen desselben wie folgt angezeigt:



Nachdem das Schema konvertiert ist, können Sie das Projekt speichern. Die Schemainformationen von der Quelldatenbank werden zusammen mit dem Projekt gespeichert. Dank dieser Funktionalität können Sie offline ohne Verbindung zur Quelldatenbank arbeiten. AWS SCT stellt eine Verbindung zur Quelldatenbank her, um das Schema in Ihrem Projekt zu aktualisieren, wenn Sie Refresh from Database (Aktualisieren von Datenbank) für Ihre Quelldatenbank wählen. Weitere Informationen finden Sie unter Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT.

Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, in dem die nicht automatisch konvertierbaren Elemente aufgeführt werden. Mit diesem Bewertungsbericht können Sie die Schemaelemente ermitteln und auflösen, die nicht automatisch konvertiert werden können. Weitere Informationen finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.

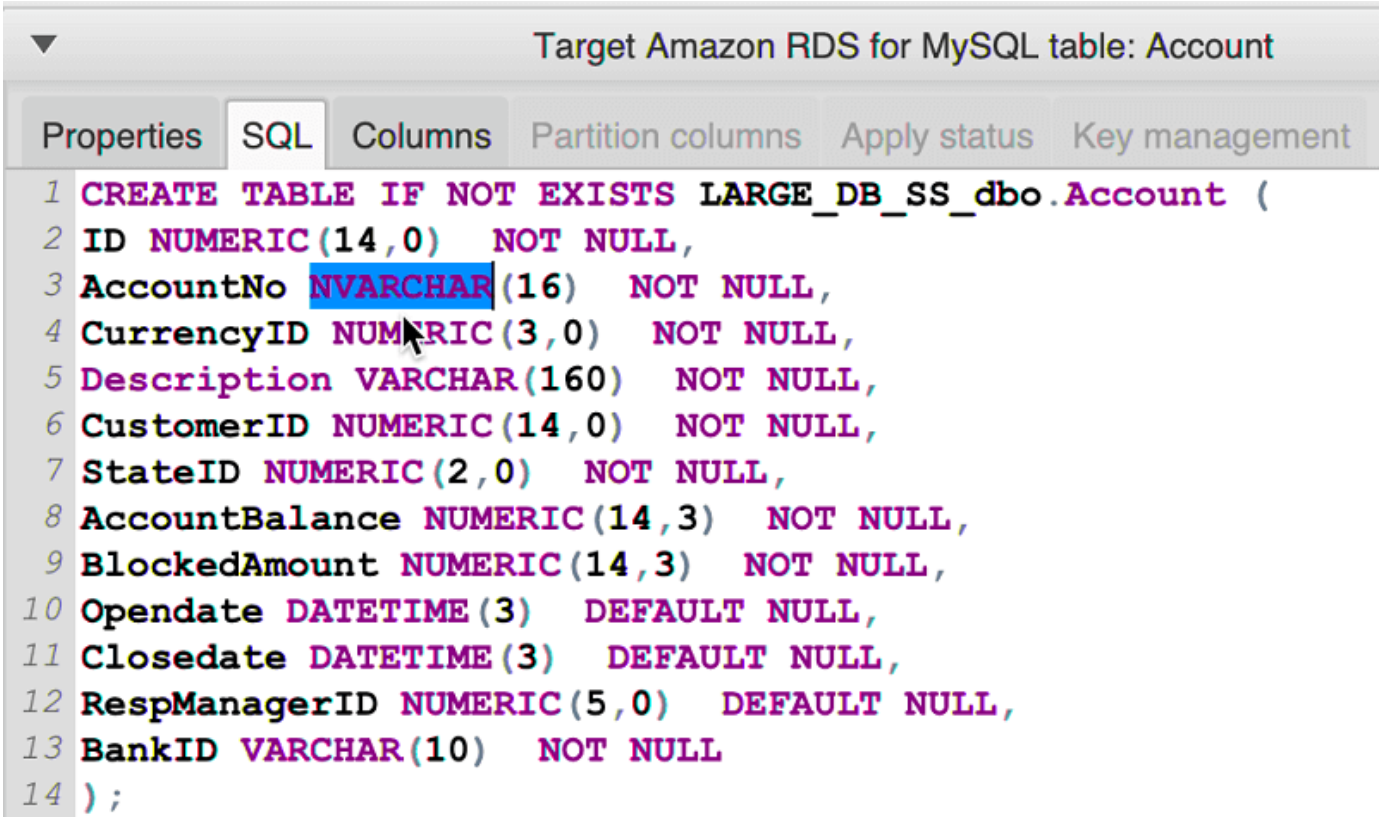
Wenn ein konvertiertes Schema AWS SCT generiert wird, wird es nicht sofort auf die Zieldatenbank angewendet. Stattdessen wird das konvertierte Schema lokal gespeichert, bis Sie es für die Zieldatenbank übernehmen. Weitere Informationen finden Sie unter Ihr konvertiertes Schema wird angewendet.

Konvertiertes Schema bearbeiten

Sie können ein konvertiertes Schema bearbeiten und die Änderungen als Teil des Projekts speichern.

So bearbeiten Sie ein konvertiertes Schema

1. Wählen Sie im linken Projektbereich, in dem das Schema von der Quelldatenbank angezeigt wird, das Schemaelement aus, für das Sie das konvertierte Schema ändern möchten.
2. Wählen Sie unten im mittleren Bereich, in dem das konvertierte Schema für das ausgewählte Element angezeigt wird, die Registerkarte SQL aus.
3. Nehmen Sie im Text, der unter der Registerkarte SQL angezeigt wird, die erforderlichen Änderungen am Schema vor. Das Schema wird automatisch für das Projekt gespeichert, wenn Sie es aktualisieren.



Die Änderungen, die Sie am konvertierten Schema vorgenommen haben, werden für das Projekt gespeichert, wenn Sie es aktualisieren. Wenn Sie ein Schemaelement aus der Quelldatenbank neu konvertieren, aber das zuvor konvertierte Schema für dieses Element bereits aktualisiert hatten, werden die erfolgten Aktualisierungen durch das auf Basis der Quelldatenbank neu konvertierte Schemaelement ersetzt.

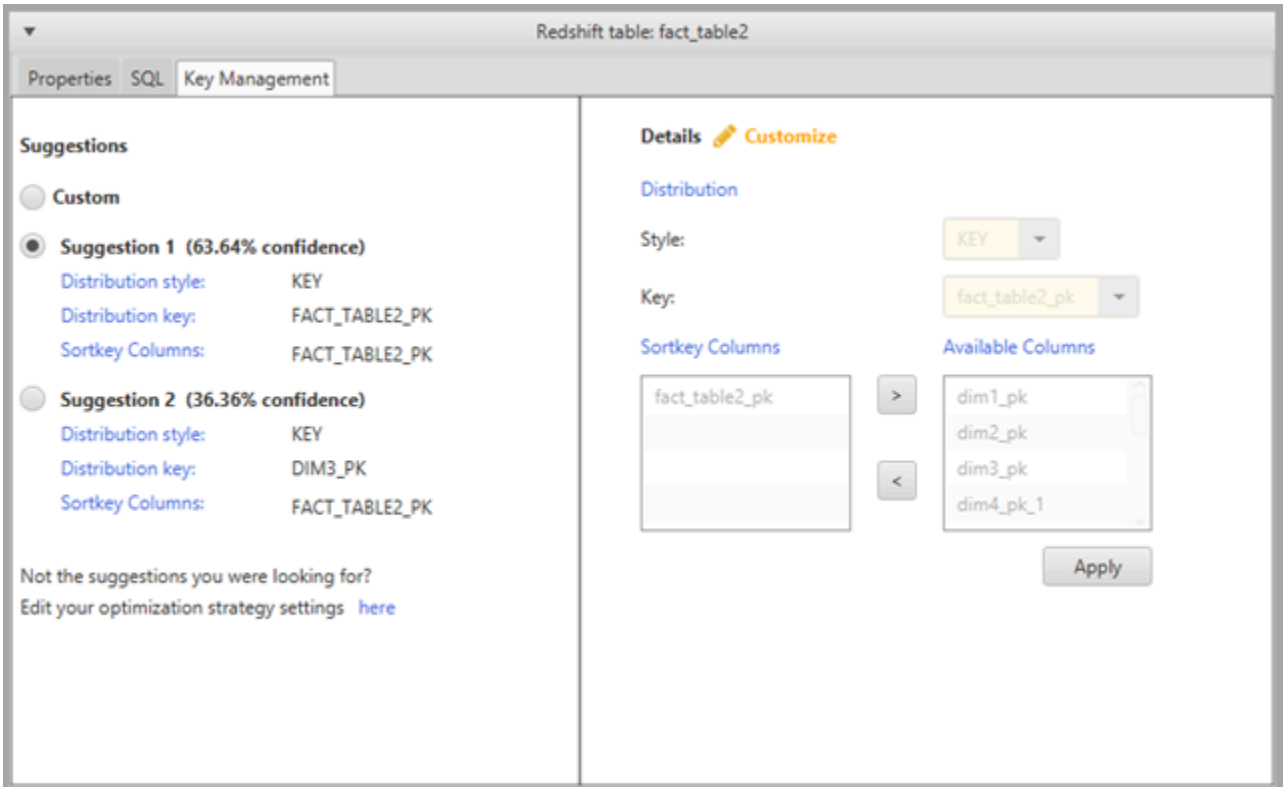
Ein konvertiertes Schema löschen

Bis Sie das Schema auf Ihre Zieldatenbank anwenden, wird das konvertierte Schema AWS SCT nur lokal in Ihrem Projekt gespeichert. Sie können das geplante Schema aus dem Projekt löschen, indem Sie in der Strukturansicht auf den Knoten für die Zieldatenbank klicken und dann Refresh from Database auswählen. Da kein Schema in Ihre Zieldatenbank geschrieben wurde, werden bei der Aktualisierung aus der Datenbank die geplanten Schemaelemente in Ihrem AWS SCT Projekt entfernt, sodass sie dem entsprechen, was in Ihrer Zieldatenbank vorhanden ist.

Schlüssel verwalten und anpassen in AWS SCT

Nachdem Sie Ihr Schema mit dem konvertiert haben AWS Schema Conversion Tool, können Sie Ihre Schlüssel verwalten und bearbeiten. Die Schlüsselverwaltung spielt eine zentrale Rolle bei der Data Warehouse-Konvertierung.

Wählen Sie zum Verwalten von Schlüsseln eine Tabelle in Ihrer Zieldatenbank und dann die Registerkarte Key Management aus, wie im Folgenden gezeigt.



Der linke Bereich enthält Schlüsselvorschläge, einschließlich einer Zuverlässigkeitsbewertung der einzelnen Vorschläge. Sie können einen der Vorschläge auswählen oder einen Schlüssel anpassen, indem Sie diesen im rechten Bereich bearbeiten.

Entspricht die Auswahl nicht Ihren Erwartungen, können Sie Ihre Optimierungsstrategien bearbeiten und erneut eine Konvertierung ausprobieren. Weitere Informationen finden Sie unter Auswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT.

Verwandte Themen

- Wählen Sie den besten Sortierschlüssel
- Wählen Sie den besten Vertriebsstil

Erstellung und Verwendung des Bewertungsberichts in AWS SCT

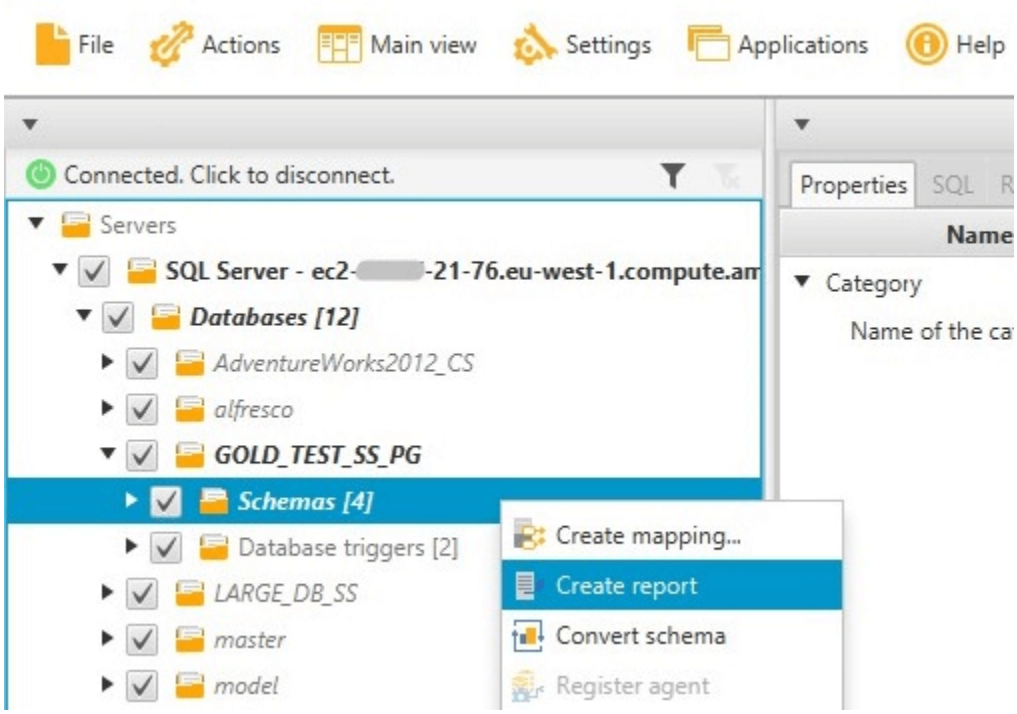
Der AWS Schema Conversion Tool erstellt einen Bewertungsbericht zur Datenbankmigration, der Sie bei der Konvertierung Ihres Schemas unterstützt. Dieser Bericht stellt wichtige Informationen zur Konvertierung des Schemas von Ihrer Quelldatenbank in die Zieldatenbank bereit. Der Bericht fasst alle Aufgaben der Schemakonvertierung zusammen und führt detailliert die Aktionselemente für Schemata auf, die nicht in die Engine der Zieldatenbank konvertiert werden können. Außerdem enthält der Bericht eine Aufwandsschätzung, da für die nicht automatisch konvertierbaren Elemente entsprechender Code in die Zieldatenbank geschrieben werden muss.

Erstellen eines Berichts zur Bewertung der Datenbankmigration

Führen Sie die folgenden Schritte aus, um einen Bewertungsbericht zur Datenbankmigration zu erstellen.

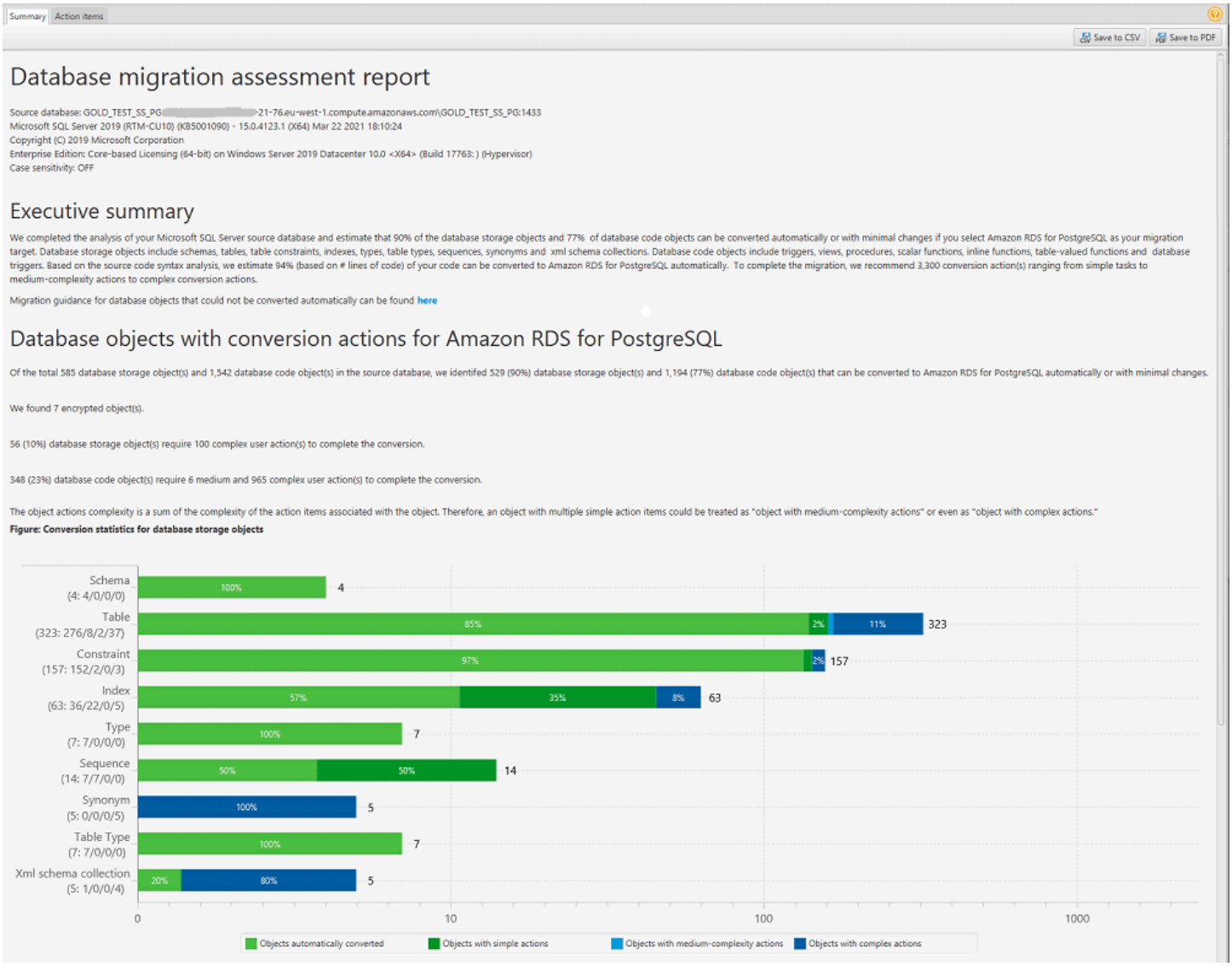
So erstellen Sie einen Bewertungsbericht zur Datenbankmigration

1. Wählen Sie im linken Bereich, in dem das Schema Ihrer Quelldatenbank angezeigt wird, ein Schemaobjekt aus, für das ein Bewertungsbericht erstellt werden soll.
2. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Create Report aus.



Zusammenfassung des Bewertungsberichts

Nach der Erstellung des Bewertungsberichts wird die entsprechende Ansicht mit der Registerkarte Summary geöffnet. Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es werden sowohl die automatisch konvertierten Elemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.



Für die Schemaelemente, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden können, enthält die Zusammenfassung eine Aufwandsschätzung zur Erstellung von Schemaelementen in Ihrer Ziel-DB-Instance, die mit denen in der Quelle gleichgesetzt werden können.

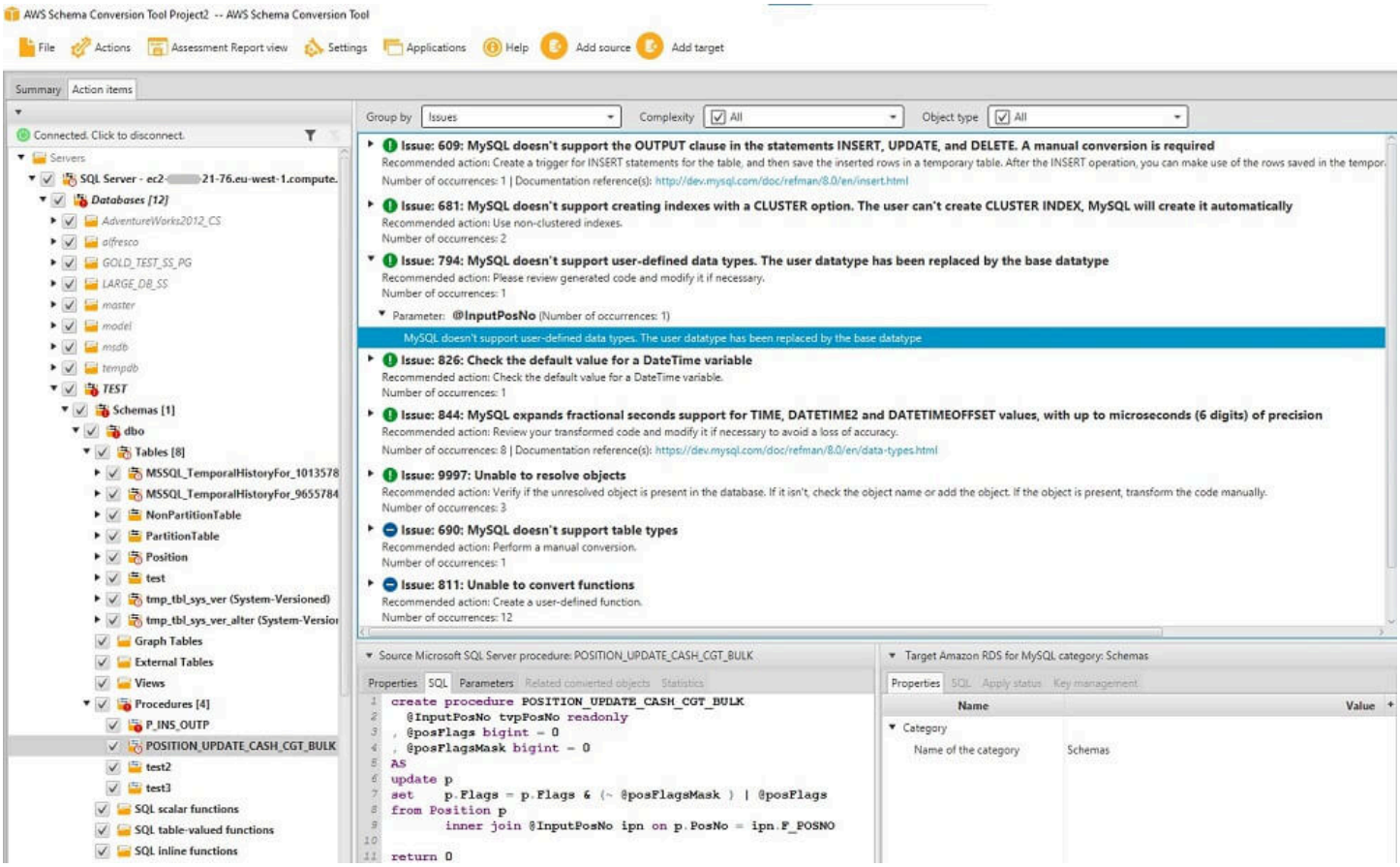
Im Bericht wird der geschätzte Zeitaufwand für die Konvertierung dieser Schemaelemente wie folgt kategorisiert:

- Einfach — Aktionen, die in weniger als einer Stunde abgeschlossen werden können.
- Mittel — Aktionen, die komplexer sind und in ein bis vier Stunden abgeschlossen werden können.
- Signifikant — Aktionen, die sehr komplex sind und mehr als vier Stunden in Anspruch nehmen.

Aktionselemente im Bewertungsbericht

Im Bewertungsbericht ist auch eine Registerkarte Action Items vorhanden. Diese Registerkarte enthält eine Liste der Elemente, die nicht automatisch in die Datenbank-Engine der Zieldatenbank konvertiert werden können. Wenn Sie ein Aktionselement aus der Liste auswählen, markiert AWS SCT das entsprechende Element im Schema, für das diese Aktion gilt.

Der Bericht enthält zudem Empfehlungen für die manuelle Konvertierung des Schemaelements. Weitere Informationen zum Festlegen der Handhabung von manuellen Konvertierungen finden Sie unter Handhabung manueller Konvertierungen in AWS SCT.



Speichern des Bewertungsberichts

Sie können eine lokale Kopie des Berichts zur Bewertung der Datenmigration entweder als PDF-Datei oder als CSV-Datei speichern. Die CSV-Datei enthält nur Informationen zu den Aktionselementen. Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

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 identifed 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.

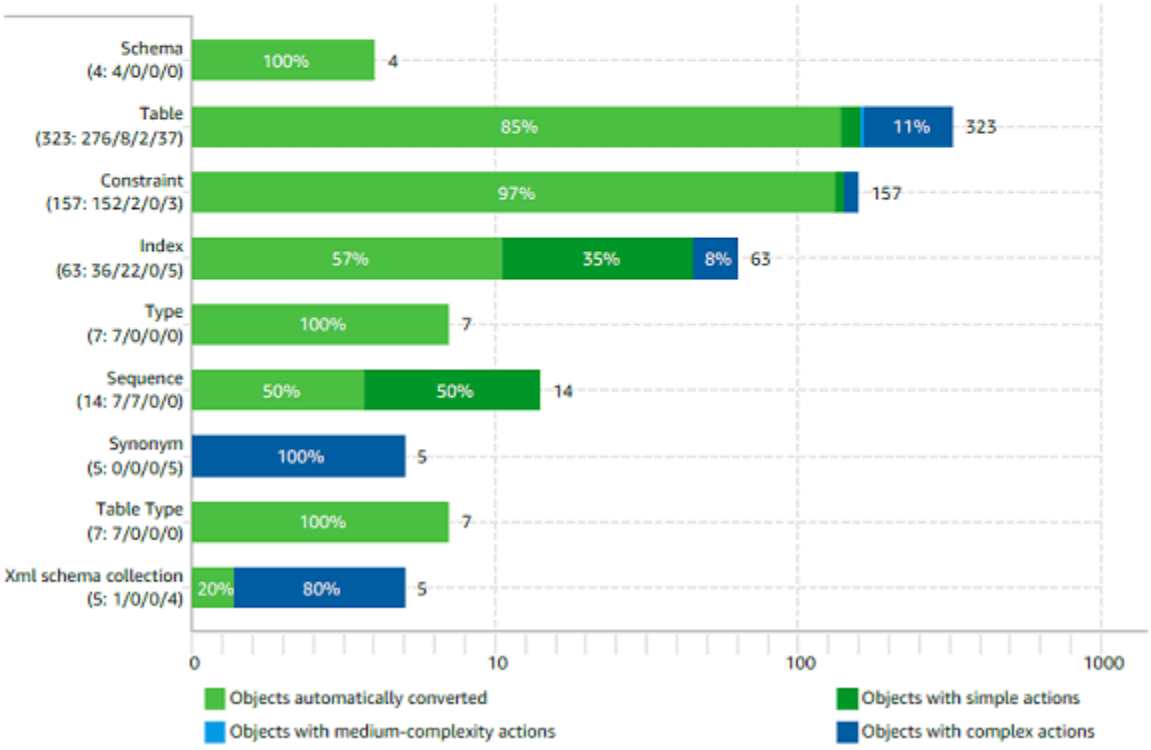
We found 7 encrypted object(s).

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



Handhabung manueller Konvertierungen in AWS SCT

Der Bewertungsbericht enthält eine Liste der Elemente, die nicht automatisch in die Datenbank-Engine der Zieldatenbank konvertiert werden können. Für jedes nicht konvertierbare Element wird auf der Registerkarte Action Items ein Aktionselement angegeben.

Sie können wie folgt auf die Aktionselemente im Bewertungsbericht reagieren:

- Sie ändern das Schema der Quelldatenbank.

- Sie ändern das Schema der Zieldatenbank.

Ihr Quellschema ändern

Bei einigen Elementen ist es einfacher, das Datenbankschema der Quelldatenbank in ein Schema zu ändern, das automatisch konvertiert werden kann. Vergewissern Sie sich zunächst, dass die neuen Änderungen mit der Anwendungsarchitektur kompatibel sind, und aktualisieren Sie dann das Schema der Quelldatenbank. Aktualisieren Sie abschließend das Projekt mit den neuen Schemainformationen. Anschließend können Sie das aktualisierte Schema konvertieren und den Bewertungsbericht zur Datenbankmigration erneut erstellen. Für die im Quellschema geänderten Elemente werden keine Aktionselemente mehr angezeigt.

Der Vorteil dieses Verfahrens besteht darin, dass das aktualisierte Schema immer verfügbar ist, wenn Sie von der Quelldatenbank aktualisieren.

Ändern Sie Ihr Zielschema

Bei einigen Elementen ist es einfacher, das konvertierte Schema für die Zieldatenbank zu übernehmen und anschließend für die Elemente, die nicht automatisch konvertiert werden konnten, entsprechende Schemaelemente manuell zur Zieldatenbank hinzuzufügen. Durch Übernehmen des jeweiligen Schemas können Sie automatisch konvertierbare Schemata in die Zieldatenbank schreiben. Weitere Informationen finden Sie unter Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT.

Das Schema, das in die Zieldatenbank geschrieben wird, enthält nur automatisch konvertierbare Elemente. Nachdem Sie das Schema für die Zieldatenbank übernommen haben, können Sie manuell ein (mit der Quelldatenbank) gleichwertiges Schema in der Zieldatenbank erstellen. Die im Bewertungsbericht zur Datenbankmigration angegebenen Aktionselemente enthalten Vorschläge zum Erstellen eines gleichwertigen Schemas.

> **Warning**
>
> Wenn Sie ein Schema manuell in der Zieldatenbank erstellen, sollten Sie eine Kopie davon speichern. Falls Sie das konvertierte Schema vom Projekt erneut in die Zieldatenbank übernehmen, werden Ihre manuellen Änderungen überschrieben.

In einigen Fällen ist es nicht möglich, ein gleichwertiges Schema in der Zieldatenbank zu erstellen. Möglicherweise muss das Design der Anwendung und der Datenbank in Teilen überarbeitet werden, damit Sie die von der Engine bereitgestellte Funktionalität für Ihre Zieldatenbank nutzen können. In anderen Fällen kann das Schema, das nicht automatisch konvertiert werden kann, ignoriert werden.

Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT

Sie können sowohl das Quellschema als auch das Zielschema in Ihrem AWS Schema Conversion Tool Projekt aktualisieren.

- Quelle — Wenn Sie das Schema für Ihre Quelldatenbank aktualisieren, AWS SCT wird das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Quelldatenbank ersetzt. Mit dieser Funktion können Sie das Projekt aktualisieren, wenn das Schema der Quelldatenbank geändert wurde.
- Ziel — Wenn Sie das Schema für Ihre Zieldatenbank aktualisieren, AWS SCT ersetzt das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Zieldatenbank. Wenn Sie kein Schema auf Ihre Zieldatenbank angewendet haben, AWS SCT löscht das konvertierte Schema aus Ihrem Projekt. Anschließend können Sie das Schema von der Quelldatenbank für eine neue Zieldatenbank konvertieren.

Sie aktualisieren das Schema in Ihrem AWS SCT Projekt, indem Sie Aus Datenbank aktualisieren wählen.

Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT

Wenn das konvertierte Schema AWS Schema Conversion Tool generiert (wie unter gezeigtKonvertieren Sie Ihr Schema mit AWS SCT), wird das konvertierte Schema nicht sofort auf die Zieldatenbank angewendet. Stattdessen wird das konvertierte Schema lokal im Projekt gespeichert, bis Sie es für die Zieldatenbank übernehmen. Mit dieser Funktion können Sie mit Schemaelementen arbeiten, die sich nicht automatisch in die Zieldatenbank-Engine konvertieren lassen. Weitere Informationen zu Elementen, die nicht automatisch konvertierbar sind, finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.

Optional können Sie das konvertierte Schema mithilfe des Tools als SQL-Skript in einer Datei speichern, bevor Sie das Schema für die Zieldatenbank übernehmen. Mit diesem Tool können Sie das konvertierte Schema auch direkt für die Zieldatenbank übernehmen.

Speichern Sie Ihr konvertiertes Schema in einer Datei

Sie können das konvertierte Schema als SQL-Skripts in einer Textdatei speichern. Mit diesem Ansatz können Sie die generierten SQL-Skripts ändern, um Elemente AWS SCT zu adressieren, die das Tool nicht automatisch konvertieren kann. Anschließend führen Sie die aktualisierten Skripts auf der DB-Ziel-Instance aus und übernehmen das konvertierte Schema in die Zieldatenbank.

So speichern Sie Ihr konvertiertes Schema als SQL-Skripts

1. Wählen Sie Ihr Schema und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie Als SQL speichern.
3. Geben Sie den Namen der Datei ein und wählen Sie Speichern.
4. Speichern Sie Ihr konvertiertes Schema mit einer der folgenden Optionen:

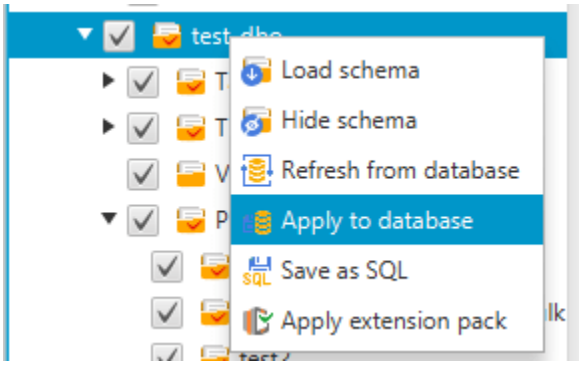
    - Einzelne Datei
    - Eine Datei pro Phase
    - Eine Datei pro Kontoauszug

Um das Format des SQL-Skripts auszuwählen

1. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen aus.
2. Wählen Sie „Skripts speichern“.
3. Wählen Sie für Vendor die Datenbankplattform aus.
4. Wählen Sie unter SQL-Skripten speichern unter aus, wie Sie Ihr Datenbankschemaskript speichern möchten.
5. Wählen Sie OK, um die Einstellungen zu speichern.

Ihr konvertiertes Schema wird angewendet

Um das konvertierte Schema für die Zieldatenbank zu übernehmen, wählen Sie im rechten Projektbereich das Schemaelement aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schemaelement und wählen Sie Apply to database aus, wie hier gezeigt.



Das Erweiterungspaket-Schema

Wenn Sie zum ersten Mal ein konvertiertes Schema für die DB-Ziel-Instance übernehmen, fügt AWS SCT ein weiteres Schema zu Ihrer DB-Ziel-Instance hinzu. Über dieses Schema werden Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Ziel-DB-Instance benötigt werden. Das Schema wird als Erweiterungspaketschema bezeichnet.

Dieses Erweiterungspaketschema sollten Sie nicht ändern, andernfalls könnten im konvertierten Schema, das auf die DB-Ziel-Instance geschrieben wird, unerwünschte Ergebnisse auftreten. Wenn Ihr Schema vollständig auf Ihre Ziel-DB-Instance migriert ist und Sie es nicht mehr benötigen AWS SCT, können Sie das Erweiterungspaket-Schema löschen.

Das Erweiterungspaketschema wird wie folgt entsprechend Ihrer Quelldatenbank benannt:

- Greenplum: `aws_greenplum_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Schneeflocke: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertica: `aws_vertica_ext`

Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Python-Bibliotheken

Um benutzerdefinierte Funktionen in Amazon Redshift zu erstellen, verwenden Sie die Sprache Python. Verwenden Sie das AWS SCT Erweiterungspaket, um Python-Bibliotheken für Ihre Amazon Redshift Redshift-Datenbank zu installieren. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

Daten aus Amazon Redshift konvertieren mit AWS Schema Conversion Tool

Sie können das verwenden AWS Schema Conversion Tool , um Ihre Amazon Redshift Redshift-Datenbank zu optimieren. Wenn Sie Ihre Amazon Redshift Redshift-Datenbank als Quelle und eine Amazon Redshift Redshift-Testdatenbank als Ziel verwenden, AWS SCT empfiehlt es sich, Schlüssel und Verteilungsschlüssel zu sortieren, um Ihre Datenbank zu optimieren.

Optimieren Sie Ihre Amazon Redshift Redshift-Datenbank

Verwenden Sie das folgende Verfahren, um Ihre Amazon Redshift Redshift-Datenbank zu optimieren.

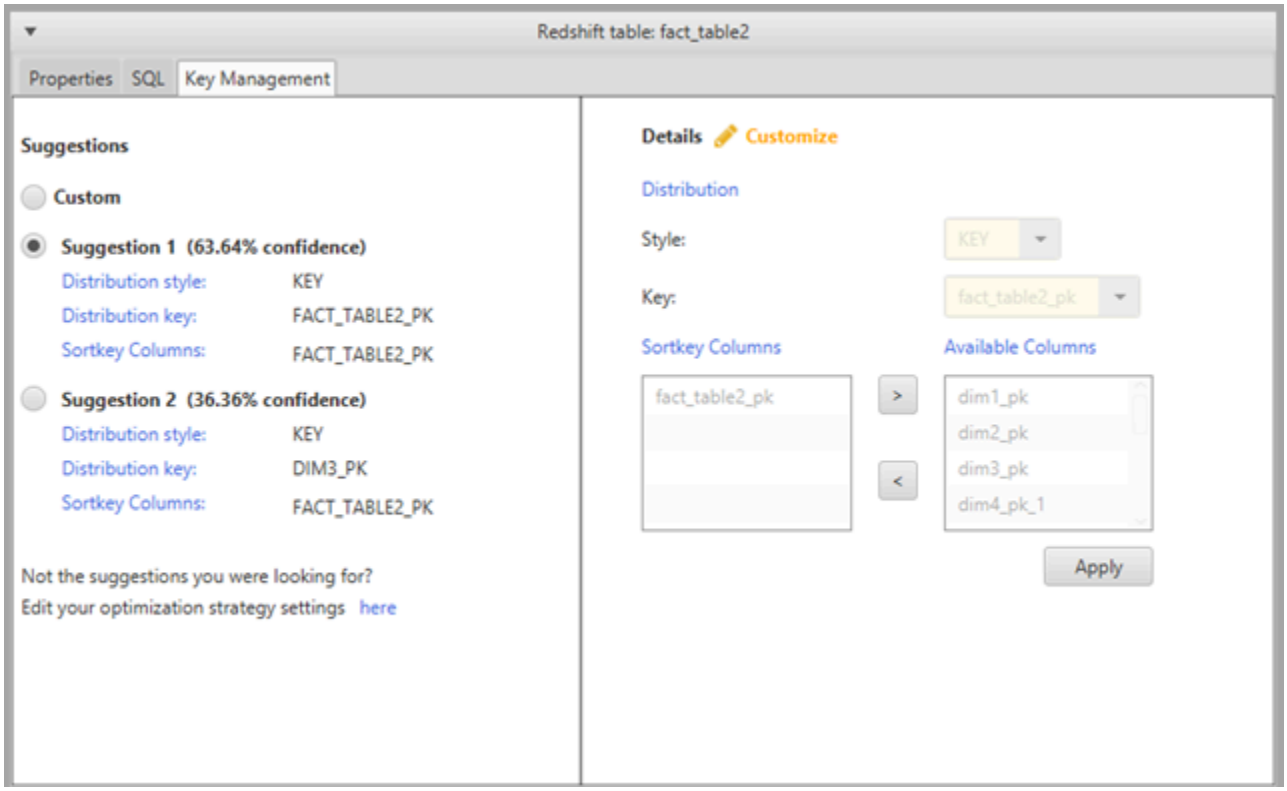
Um Ihre Amazon Redshift Redshift-Datenbank zu optimieren

1. Erstellen Sie einen manuellen Snapshot Ihres Amazon Redshift Redshift-Clusters als Backup. Sie können den Snapshot löschen, nachdem Sie Ihren Amazon Redshift Redshift-Cluster optimiert und alle von Ihnen vorgenommenen Änderungen getestet haben. Weitere Informationen finden Sie unter Amazon Redshift Redshift-Snapshots.
2. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie konvertieren möchten. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Collect Statistics aus.

   AWS SCT verwendet die Statistiken, um Vorschläge für Sortier- und Verteilungsschlüssel zu machen.
3. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie optimieren möchten. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Run Optimization aus.

   AWS SCT macht Vorschläge für Sortierschlüssel und Verteilungsschlüssel.

4. Um diese Empfehlungen zu prüfen, erweitern Sie den Tabellenknoten unter dem Schema im linken Bereich des Projekts und wählen Sie eine Tabelle aus. Klicken Sie auf die Registerkarte Key Management, wie hier gezeigt.



   Der linke Bereich enthält Schlüsselvorschläge, einschließlich einer Zuverlässigkeitsbewertung der einzelnen Vorschläge. Sie können einen der Vorschläge auswählen oder einen Schlüssel anpassen, indem Sie diesen im rechten Bereich bearbeiten.

5. Sie können einen Bericht mit den Optimierungsempfehlungen erstellen. Gehen Sie wie folgt vor, um den Bericht zu erstellen:

   a. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie optimiert haben. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Create Report aus.

      Der Bericht wird im Hauptfenster geöffnet und die Registerkarte Summary wird angezeigt. Die Anzahl der Objekte mit Optimierungsempfehlungen wird im Bericht angezeigt.

   b. Klicken Sie auf die Registerkarte Action Items, um die wichtigsten Empfehlungen in einem Berichtsformat anzuzeigen.

   c. Sie können eine lokale Kopie des Optimierungsberichts entweder als PDF-Datei oder als CSV-Datei speichern. Die CSV-Datei enthält nur Informationen zu den Aktionselementen.

Die PDF-Datei enthält sowohl die Zusammenfassung als auch Informationen zu den Aktionselementen.

6. Um die empfohlenen Optimierungen auf Ihre Datenbank anzuwenden, wählen Sie im rechten Bereich Ihres Projekts ein Objekt aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Apply to database aus.

Konvertieren von Daten mithilfe von ETL-Prozessen in AWS Schema Conversion Tool

Sie können AWS Schema Conversion Tool (AWS SCT) verwenden, um Prozesse zum Extrahieren, Transformieren und Laden (ETL) zu migrieren. Diese Art der Migration beinhaltet die Konvertierung der ETL-bezogenen Geschäftslogik. Diese Logik kann sich entweder in Ihren Quell-Data Warehouses oder in externen Skripten befinden, die Sie separat ausführen.

AWS SCT Unterstützt derzeit die Konvertierung von ETL-Skripts in Objekte in AWS Glue und Amazon Redshift RSQL, wie in der folgenden Tabelle dargestellt.

Quelle	Ziel
ETL-Skripts von Informatica	Informatica
ETL-Pakete für Microsoft SQL Server Integration Services (SSIS)	AWS Glue oder AWS Glue Studio
Shell-Skripts mit eingebetteten Befehlen aus Teradata Basic Teradata Query (BTEQ)	Amazon-Redshift-RSQL
Teradata BTEQ ETL-Skripts	AWS Glue oder Amazon Redshift RSQL
FastExport Teradata-Jobskripte	Amazon-Redshift-RSQL
FastLoad Teradata-Jobskripte	Amazon-Redshift-RSQL
MultiLoad Teradata-Jobskripte	Amazon-Redshift-RSQL

Themen

- Konvertierung von ETL-Prozessen in AWS GlueAWS Schema Conversion Tool
- Konvertierung von ETL-Prozessen AWS Glue in AWS Schema Conversion Tool
- Konvertierung von Informatica ETL-Skripten mit AWS Schema Conversion Tool
- SSIS wird in mit konvertiert AWS GlueAWS SCT
- SSIS-Pakete werden in with konvertiert AWS Glue StudioAWS Schema Conversion Tool

- Konvertieren von Teradata BTEQ-Skripten nach Amazon Redshift RSQL mit AWS SCT
- Konvertierung von Shell-Skripten mit eingebetteten BTEQ-Befehlen nach Amazon Redshift RSQL mit AWS Schema Conversion Tool
- Konvertieren von FastExport Skripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool
- Konvertieren von FastLoad Jobskripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool
- Konvertieren von MultiLoad Skripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool

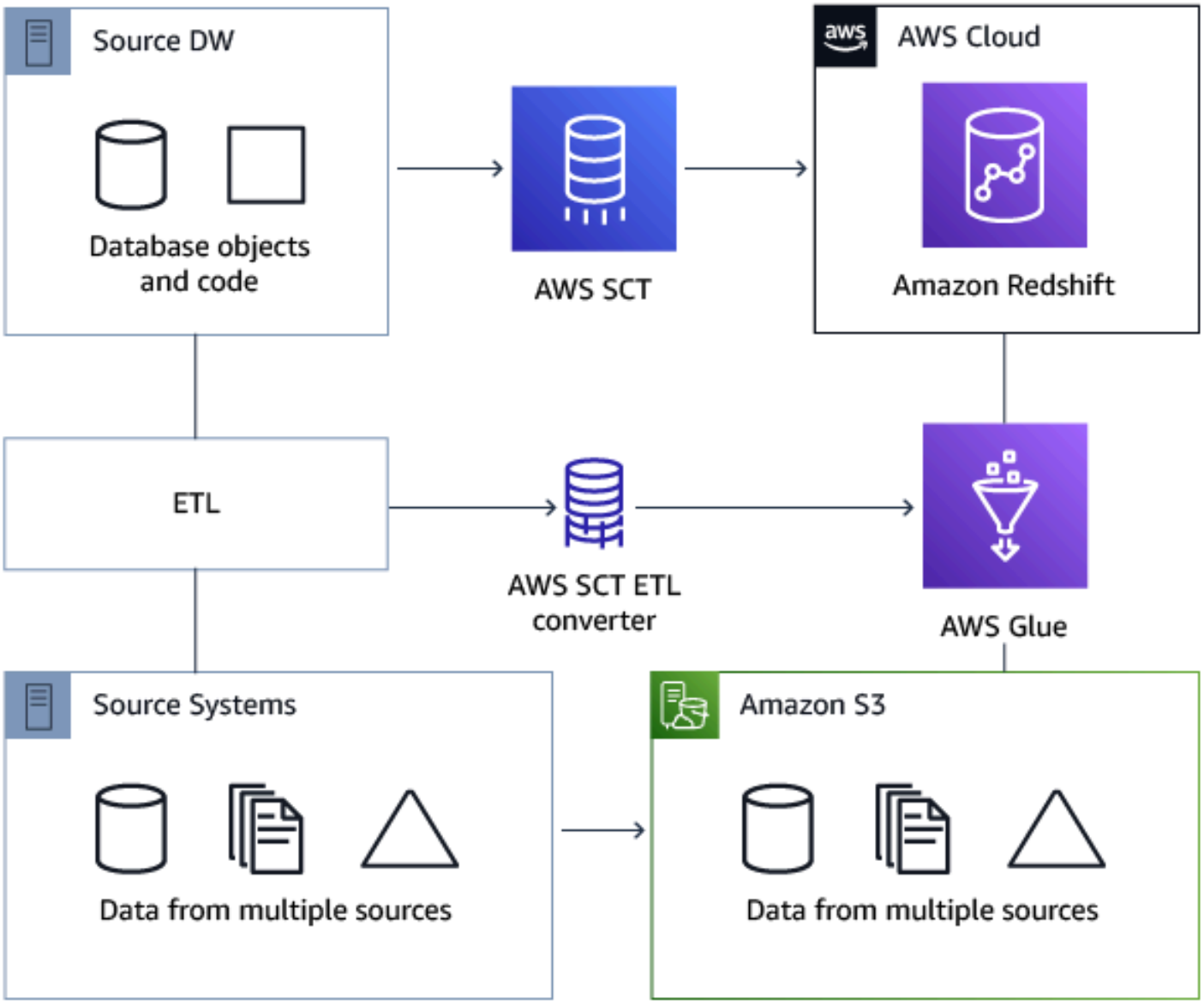
Konvertierung von ETL-Prozessen in AWS GlueAWS Schema Conversion Tool

Im Folgenden finden Sie einen Überblick über den Prozess zur Konvertierung von ETL-Skripten in AWS Glue WITH AWS SCT. In diesem Beispiel konvertieren wir eine Oracle-Datenbank zusammen mit den ETL-Prozessen, die mit den Quelldatenbanken und Data Warehouses verwendet werden, in Amazon Redshift.

Themen

- Voraussetzungen
- Den Datenkatalog verstehen AWS Glue
- Einschränkungen bei der Konvertierung AWS SCT mithilfe von AWS Glue
- Schritt 1: Erstellen eines neuen -Projekts
- Schritt 2: Erstellen Sie einen AWS Glue Job

Das folgende Architekturdiagramm zeigt ein Beispiel für ein Datenbankmigrationsprojekt, das die Konvertierung von ETL-Skripts in AWS Glue beinhaltet.



Voraussetzungen

Bevor Sie beginnen, führen Sie die folgenden Schritte aus:

- Migrieren Sie alle Quelldatenbanken, zu denen Sie migrieren möchten AWS.
- Migrieren Sie die Ziel-Data Warehouses zu AWS.
- Erfassen Sie eine Liste des gesamten Codes Ihres ETL-Prozesses.
- Erstellen Sie eine Liste aller erforderlichen Verbindungsinformationen für jede Datenbank.

AWS Glue Benötigt außerdem Berechtigungen für den Zugriff auf andere AWS Ressourcen in Ihrem Namen. Sie stellen diese Berechtigungen mithilfe von AWS Identity and Access Management (IAM) bereit. Stellen Sie sicher, dass Sie eine IAM-Richtlinie für erstellt haben. AWS GlueWeitere Informationen finden Sie unter Erstellen einer IAM-Richtlinie für den AWS Glueservice im AWS Glue Entwicklerhandbuch.

Den Datenkatalog verstehen AWS Glue

AWS Glue Lädt im Rahmen des Konvertierungsprozesses Informationen zu den Quell- und Zieldatenbanken. Es organisiert diese Informationen in Kategorien in einer Struktur, die als Baum bezeichnet wird. Diese Struktur umfasst Folgendes:

- Verbindungen — Verbindungsparameter
- Crawler — Eine Liste von Crawlern, ein Crawler für jedes Schema
- Datenbanken — Container, die Tabellen enthalten
- Tabellen — Metadatendefinitionen, die die Daten in den Tabellen darstellen
- ETL-Jobs — Geschäftslogik, die die ETL-Arbeit ausführt
- Trigger — Logik, die steuert, wann ein ETL-Job ausgeführt wird AWS Glue (ob bei Bedarf, nach Zeitplan oder ausgelöst durch Job-Ereignisse)

Der AWS Glue -Datenkatalog ist ein Index für die Speicherort-, Schema- und Laufzeitmetriken Ihrer Daten. Wenn Sie mit AWS Glue und arbeiten AWS SCT, enthält der AWS Glue Datenkatalog Verweise auf Daten, die als Quellen und Ziele Ihrer ETL-Jobs in verwendet werden AWS Glue. Um Ihr Data Warehouse zu erstellen, katalogisieren Sie diese Daten.

Sie verwenden die Informationen im Datenkatalog, um Ihre ETL-Aufträge zu erstellen und zu überwachen. Normalerweise führen Sie einen Crawler aus, um die Daten in Ihren Datenspeichern zu inventarisieren. Es gibt jedoch auch andere Möglichkeiten, Metadatentabellen in Ihren Datenkatalog einzufügen.

Wenn Sie eine Tabelle in Ihrem Datenkatalog definieren, fügen Sie sie einer Datenbank hinzu. Eine Datenbank wird verwendet, um Tabellen zu organisieren AWS Glue.

Einschränkungen bei der Konvertierung AWS SCT mithilfe von AWS Glue

Die folgenden Einschränkungen gelten für die Konvertierung AWS SCT mit AWS Glue.

Ressource	Standardlimit
Anzahl der Datenbanken pro Konto	10.000
Anzahl der Tabellen pro Datenbank	100 000
Anzahl der Partitionen pro Tabelle	1 000 000
Anzahl der Tabellenversionen für jede Tabelle	100 000
Anzahl der Tabellen für jedes Konto	1 000 000
Anzahl der Partitionen für jedes Konto	10 000 000
Anzahl der Tabellenversionen für jedes Konto	1 000 000
Anzahl der Verbindungen für jedes Konto	1.000
Anzahl der Crawler für jedes Konto	25
Anzahl der Aufträge pro Konto	25
Anzahl der Auslöser für jedes Konto	25
Anzahl der gleichzeitigen Aufträge pro Konto	30
Anzahl der gleichzeitigen Auftragsausführungen für jeden Auftrag	3
Anzahl der Aufträge für jeden Auslöser	10
Anzahl der Entwicklungsendpunkte für jedes Konto	5
Maximale Anzahl von Datenverarbeitungseinheiten (DPUs), die von einem Entwicklungsendpunkt gleichzeitig verwendet werden	5
DPUs Höchstzahl, die von einer Rolle gleichzeitig verwendet wird	100

Länge des Datenbanknamens	Unbegrenzt Aus Kompatibilitätsgründen mit anderen Metadatenspeichern, wie z. B. Apache Hive, wird der Name in Kleinbuchstaben gespeichert. Wenn Sie von Amazon Athena aus auf die Datenbank zugreifen möchten, geben Sie einen Namen an, der nur alphanumerische Zeichen und Unterstriche enthält.
Länge des Verbindungsnamens	Unbegrenzt
Länge des Crawler-Namens	Unbegrenzt

Schritt 1: Erstellen eines neuen -Projekts

Gehen Sie wie folgt vor, um ein neues Projekt zu erstellen:

1. Erstellen Sie ein neues Projekt in AWS SCT. Weitere Informationen finden Sie unter Projekte starten und verwalten in AWS SCT.
2. Fügen Sie dem Projekt Ihre Quell- und Zieldatenbanken hinzu. Weitere Informationen finden Sie unter Server zum Projekt hinzufügen in AWS SCT.

   Vergewissern Sie sich, dass Sie AWS Glue in den Verbindungseinstellungen für die Zieldatenbank die Option Verwenden ausgewählt haben. Wählen Sie dazu die AWS GlueRegisterkarte. Wählen Sie unter Aus AWS Profil kopieren das Profil aus, das Sie verwenden möchten. Das Profil sollte den AWS Zugriffsschlüssel, den geheimen Schlüssel und den Amazon S3 S3-Bucket-Ordner automatisch ausfüllen. Wenn dies nicht der Fall ist, geben Sie diese Informationen selbst ein. Nachdem Sie OK ausgewählt haben, AWS Glue werden die Objekte analysiert und Metadaten in den AWS Glue Datenkatalog geladen.

   Je nach Sicherheitseinstellungen erhalten Sie möglicherweise eine Warnmeldung, die Sie darüber informiert, dass Ihr Konto nicht über ausreichende Berechtigungen für einige der Schemata auf dem Server verfügt. Wenn Sie Zugriff auf die verwendeten Schemata haben, können Sie diese Meldung ignorieren.

3. Um die Vorbereitung des ETL-Imports abzuschließen, stellen Sie eine Verbindung zu Ihren Quell- und Zieldatenbanken her. Wählen Sie dazu Ihre Datenbank in der Quell- oder Ziel-Metadatenstruktur aus und wählen Sie dann Mit dem Server verbinden aus.

AWS Glue erstellt eine Datenbank auf dem Quelldatenbankserver und eine auf dem Zieldatenbankserver, um die ETL-Konvertierung zu unterstützen. Die Datenbank auf dem Zielserver enthält den AWS Glue Datenkatalog. Verwenden Sie die Suche im Quell- oder Zielfenster, um nach bestimmten Objekten zu suchen.

Um zu sehen, wie ein bestimmtes Objekt konvertiert wird, suchen Sie nach einem Element, das Sie konvertieren möchten, und wählen Sie im Kontextmenü (Rechtsklick) die Option Schema konvertieren. AWS SCT wandelt dieses ausgewählte Objekt in ein Skript um.

Sie können das konvertierte Skript im Ordner Scripts im rechten Bereich überprüfen. Derzeit ist das Skript ein virtuelles Objekt, das nur als Teil Ihres AWS SCT Projekts verfügbar ist.

Um einen AWS Glue Job mit Ihrem konvertierten Skript zu erstellen, laden Sie Ihr Skript auf Amazon S3 hoch. Um das Skript auf Amazon S3 hochzuladen, wählen Sie das Skript und dann im Kontextmenü (Rechtsklick) die Option In S3 speichern aus.

Schritt 2: Erstellen Sie einen AWS Glue Job

Nachdem Sie das Skript in Amazon S3 gespeichert haben, können Sie es auswählen und dann AWS Glue Job konfigurieren wählen, um den Assistenten zur Konfiguration des AWS Glue Jobs zu öffnen. Der Assistent erleichtert die Einrichtung:

1. Auf der ersten Registerkarte des Assistenten, dem Design Data Flow, können Sie eine Ausführungsstrategie und die Liste der Skripts auswählen, die Sie in diesen einen Job aufnehmen möchten. Sie können für jedes Skript Parameter auswählen. Sie können die Skripts auch neu anordnen, damit sie in der richtigen Reihenfolge ausgeführt werden.
2. Auf der zweiten Registerkarte können Sie Ihrem Job einen Namen geben und die Einstellungen für ihn direkt konfigurieren AWS Glue. Auf diesem Bildschirm können Sie die folgenden Einstellungen vornehmen:
   - AWS Identity and Access Management (IAM) -Rolle
   - Skript-Dateinamen und -Dateipfade
   - Verschlüsseln Sie das Skript mithilfe serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3)

- Temporäres Verzeichnis
- Generierter Python-Bibliothekspfad
- Python-Bibliothekspfad des Benutzers
- Pfad für die abhängigen .jar-Dateien
- Pfad für referenzierte Dateien
- Gleichzeitig DPUs für jeden ausgeführten Job
- Maximale Gleichzeitigkeit
- Auftrag-Zeitüberschreitung (in Minuten)
- Schwellenwert für die Verzögerungsbenachrichtigung (in Minuten)
- Anzahl der Wiederholungen
- Security configuration (Sicherheitskonfiguration)
- Server-side encryption

3. Im dritten Schritt bzw. auf der dritten Registerkarte wählen Sie die konfigurierte Verbindung zum Zielendpunkt aus.

Nachdem Sie die Konfiguration des Jobs abgeschlossen haben, wird er unter den ETL-Jobs im AWS Glue Datenkatalog angezeigt. Wenn Sie den Auftrag auswählen, werden die zugehörigen Einstellungen angezeigt, sodass Sie sie überprüfen oder bearbeiten können. Um einen neuen Job in zu erstellen AWS Glue AWS Glue , wählen Sie im Kontextmenü (Rechtsklick) für den Job die Option Job erstellen. Dadurch wird die Schemadefinition angewendet. Um die Anzeige zu aktualisieren, wählen Sie Refresh from database (Aus der Datenbank aktualisieren) aus dem Kontextmenü.

Zu diesem Zeitpunkt können Sie Ihren Job in der AWS Glue Konsole anzeigen. Melden Sie sich dazu bei der an AWS Management Console und öffnen Sie die AWS Glue Konsole unter https://console.aws.amazon.com/glue/.

Sie können den neuen Auftrag testen, um sicherzustellen, dass er korrekt funktioniert. Überprüfen Sie die Daten in Ihrer Quelltabelle und prüfen Sie dann, ob die Zieltabelle leer ist. Führen Sie den Auftrag aus und überprüfen Sie erneut. Sie können Fehlerprotokolle von der AWS Glue Konsole aus einsehen.

Konvertierung von ETL-Prozessen AWS Glue in AWS Schema Conversion Tool

In den folgenden Abschnitten finden Sie eine Beschreibung einer Konvertierung, die AWS Glue API-Operationen in Python aufruft. Weitere Informationen finden Sie unter Programmieren von AWS Glue ETL-Skripts in Python im AWS Glue Developer Guide.

Themen

- Schritt 1: Erstellen einer Datenbank
- Schritt 2: Stellen Sie eine Verbindung her
- Schritt 3: Erstellen Sie einen AWS Glue Crawler

Schritt 1: Erstellen einer Datenbank

Der erste Schritt besteht darin, mithilfe der AWS SDK-API eine neue Datenbank in einem AWS Glue Datenkatalog zu erstellen. Wenn Sie eine Tabelle im Datenkatalog definieren, fügen Sie sie einer Datenbank hinzu. Eine Datenbank wird verwendet, um die Tabellen zu organisieren AWS Glue.

Das folgende Beispiel demonstriert die `create_database` Methode der Python-API für AWS Glue.

```
response = client.create_database(
    DatabaseInput={
        'Name': 'database_name',
        'Description': 'description',
        'LocationUri': 'string',
        'Parameters': {
            'parameter-name': 'parameter value'
        }
    }
)
```

Wenn Sie Amazon Redshift verwenden, wird der Datenbankname wie folgt gebildet.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Der vollständige Name des Amazon Redshift Redshift-Clusters für dieses Beispiel lautet wie folgt.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Die folgende Abbildung zeigt ein Beispiel für einen gültigen Datenbanknamen. In diesem Fall ist `rsdbb03` der Name, bei dem es sich um den ersten Teil des vollständigen Namens des Cluster-Endpunkts handelt. Die Datenbank trägt den Namen `dev` und das Schema ist `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Schritt 2: Stellen Sie eine Verbindung her

Erstellen Sie mithilfe der AWS SDK-API eine neue Verbindung in einem Datenkatalog.

Das folgende Beispiel zeigt die Verwendung der `create_connection`Methode der Python-API für 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'
            ]
        }
    }
)
```

In `create_connection` werden die folgenden Parameter verwendet:

- `Name`(UTF-8-Zeichenfolge) — erforderlich. Für Amazon Redshift wird der Verbindungsname wie folgt gebildet:`Redshift_<Endpoint-name>_<redshift-database-name>`, zum Beispiel: `Redshift_abcde03_dev`
- `Description`(UTF-8-Zeichenfolge) — Ihre Beschreibung der Verbindung.

- `ConnectionType`(UTF-8-Zeichenfolge) — Erforderlich. Die Art der Verbindung. Derzeit wird nur JDBC unterstützt, SFTP wird nicht unterstützt.
- `ConnectionProperties`(dict) — Erforderlich. Eine Liste von Schlüssel-Wert-Paaren, die als Parameter für diese Verbindung verwendet werden, einschließlich der JDBC-Verbindungs-URL, des Benutzernamens und des Kennworts.
- `PhysicalConnectionRequirements`(dict) — Physikalische Verbindungsanforderungen, die Folgendes beinhalten:
  - `SubnetId`(UTF-8-Zeichenfolge) — Die ID des Subnetzes, das von der Verbindung verwendet wird.
  - `SecurityGroupIdList`(Liste) — Die von der Verbindung verwendete Sicherheitsgruppen-ID-Liste.
  - `AvailabilityZone`(UTF-8-Zeichenfolge) — Erforderlich. Die Availability Zone, die den Endpunkt enthält. Dieser Parameter ist veraltet.

Schritt 3: Erstellen Sie einen AWS Glue Crawler

Als Nächstes erstellen Sie einen AWS Glue Crawler, um den Katalog zu füllen. AWS Glue Weitere Informationen finden Sie unter Katalogisieren von Tabellen mit einem Crawler im Entwicklerhandbuch.AWS Glue

Der erste Schritt beim Hinzufügen eines Crawlers besteht darin, mithilfe der SDK-API eine neue Datenbank in einem Datenkatalog zu erstellen.AWS Bevor Sie beginnen, stellen Sie sicher, dass Sie zunächst alle früheren Versionen löschen, indem Sie den `delete_crawler` Vorgang ausführen.

Beim Erstellen des Crawlers gilt es einige Punkte zu berücksichtigen:

- Verwenden Sie für Crawler-Namen das Format *`<redshift_node_name>`*`_`*`<redshift_database_name>`*`_`*`<redshift_shema_name>`*, z. B.: `abcde03_dev_ora_glue`
- Verwenden Sie eine IAM-Rolle, die bereits vorhanden ist. Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter Erstellen von IAM-Rollen im IAM-Benutzerhandbuch.
- Verwenden Sie den Namen der Datenbank, die Sie im vorherigen Schritt erstellt haben.
- Verwenden Sie den erforderlichen Parameter `ConnectionName`.
- Verwenden Sie für den Parameter `path` den Pfad des JDBC-Ziels, z. B.: `dev/ora_glue/%`

Im folgenden Beispiel wird ein vorhandener Crawler gelöscht und dann wird ein neuer Crawler mit der Python-API für AWS Glue erstellt:

```
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'
)
```

Nach der Crawler-Erstellung führen Sie einen Crawler aus, der eine Verbindung zu einem oder mehreren Datenspeichern herstellt, die Datenstrukturen bestimmt und Tabellen in den Datenkatalog schreibt. Sie können Ihren Crawler nach einem Zeitplan ausführen, wie im Folgenden gezeigt.

```
response = client.start_crawler(
    Name='string'
)
```

In diesem Beispiel wird Amazon Redshift als Ziel verwendet. Amazon Redshift Redshift-Datentypen AWS Glue werden Datentypen auf folgende Weise zugeordnet, nachdem der Crawler ausgeführt wurde.

Amazon Redshift Redshift-Datentyp	AWS Glue Datentyp
smallint	smallint
Ganzzahl	int
bigint	bigint
Dezimalwert	decimal(18,0)
decimal(p,s)	decimal(p,s)
real	double
double precision	double
Boolean	Boolean
char	string
varchar	string
varchar(n)	string
date	date
timestamp	timestamp
timestamptz	Zeitstempel

Konvertierung von Informatica ETL-Skripten mit AWS Schema Conversion Tool

Sie können die AWS SCT Befehlszeilenschnittstelle (CLI) verwenden, um Ihre Informatica ETL-Skripts zu konvertieren, sodass Sie die Skripts mit Ihrer neuen Zieldatenbank verwenden können. Diese Konvertierung umfasst drei wichtige Schritte. AWS SCT Konvertiert zunächst den SQL-Code, der in Ihre Informatica-Objekte eingebettet ist. Als Nächstes werden die Namen der Datenbankobjekte entsprechend den Migrationsregeln AWS SCT geändert, die Sie in Ihrem Projekt angegeben haben. Leitet AWS SCT abschließend die Verbindungen Ihrer Informatica ETL-Skripts zur neuen Zieldatenbank um.

Sie können Informatica-ETL-Skripts im Rahmen eines AWS SCT Datenbankkonvertierungsprojekts konvertieren. Stellen Sie sicher, dass Sie Ihre Quell- und Zieldatenbanken zum Projekt hinzufügen, wenn Sie Informatica ETL-Skripts konvertieren.

Um Informatica ETL-Skripts zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.667 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle von vertraut. AWS SCT Weitere Informationen finden Sie unter CLI-Referenz für AWS Schema Conversion Tool.

Um Informatica ETL-Skripts zu konvertieren mit AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können die `InformaticConversionTemplate.scts` Vorlage beispielsweise herunterladen und bearbeiten. Weitere Informationen finden Sie unter CLI-Szenarien abrufen.
2. Laden Sie die erforderlichen JDBC-Treiber für Ihre Quell- und Zieldatenbanken herunter. Geben Sie den Speicherort dieser Treiber mit dem Befehl an`SetGlobalSettings`. Geben Sie außerdem die Ordner an, in denen Protokolldateien gespeichert werden AWS SCT können.

   Das folgende Codebeispiel zeigt Ihnen, wie Sie den Pfad zu den Oracle- und PostgreSQL-Treibern zu den AWS SCT Einstellungen hinzufügen. AWS SCT Speichert Protokolldateien im Ordner, nachdem Sie dieses Codebeispiel ausgeführt haben. `C:\sct_log` AWS SCT Speichert außerdem Konsolen-Protokolldateien im `C:\Temp\oracle_postgresql` Ordner.

```
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. Erstellen Sie ein neues AWS SCT Projekt. Geben Sie den Namen und den Speicherort Ihres Projekts ein.

   Im folgenden Codebeispiel wird das `oracle_postgresql` Projekt im `C:\Temp` Ordner erstellt.

```
CreateProject
 -name: 'oracle_postgresql'
 -directory: 'C:\Temp'
/
```

4. Fügen Sie Verbindungsinformationen zu Ihren Quell- und Zieldatenbanken hinzu.

   Das folgende Codebeispiel fügt Oracle- und PostgreSQL-Datenbanken als Quelle und Ziel für Ihr AWS SCT Projekt hinzu.

```
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'
/
```

Ersetzen Sie im vorherigen Beispiel *source_user* und *target_user* durch die Namen Ihrer Datenbankbenutzer. Ersetzen Sie als Nächstes *source_password* und *target_password* durch Ihre Passwörter. Geben Sie für *source_address* und *target_address* die IP-Adressen Ihrer Quell- und Zieldatenbankserver ein.

Um eine Verbindung zu einer Oracle-Datenbank der Version 19 und höher herzustellen, verwenden Sie den Oracle-Dienstnamen im `AddSource` Befehl. Fügen Sie dazu den `-connectionType` Parameter hinzu und setzen Sie seinen Wert auf`'basic_service_name'`. Fügen Sie dann den `-servicename` Parameter hinzu und setzen Sie seinen Wert auf Ihren Oracle-Dienstnamen. Weitere Informationen zu dem `AddSource` Befehl finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

5. Erstellen Sie eine neue AWS SCT Zuordnungsregel, die die Zieldatenbank-Engines für jedes Quelldatenbankschema definiert. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.

   Das folgende Codebeispiel erstellt eine Zuordnungsregel, die alle Oracle-Quelldatenbankschemas umfasst und PostgreSQL als Migrationsziel definiert.

```
AddServerMapping
 -sourceTreePath: 'Servers.ORACLE'
 -targetTreePath: 'Servers.POSTGRESQL'
/
```

6. Fügen Sie Verbindungsinformationen zu Ihren Informatica-Quell- und Ziel-XML-Dateien hinzu.

   Im folgenden Codebeispiel werden die Informatica-XML-Dateien aus den Ordnern `C:\Informatica_source` und `C:\Informatica_target` hinzugefügt.

```
AddSource
 -name: 'INFA_SOURCE'
 -vendor: 'INFORMATICA'
 -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
 -name: 'INFA_TARGET'
 -vendor: 'INFORMATICA'
 -mappingsFolder: 'C:\Informatica_target'
/
```

7. Erstellen Sie eine weitere Zuordnungsregel, um die Informatica XML-Zieldatei für Ihre Informatica XML-Quelldatei zu definieren.

   Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die die im vorherigen Beispiel verwendeten Informatica-XML-Quelldateien und die Zieldateien umfasst.

```
AddServerMapping
-sourceTreePath: 'ETL.INFA_SOURCE'
-targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Geben Sie die Datenbankserver-Verbindung an, die der Referenz zum Informatica-Verbindungsnamen entspricht.

   Das folgende Codebeispiel konfiguriert die Umleitung Ihrer Informatica-ETL-Skripts von Ihrer Quell- zur neuen Zieldatenbank. In diesem Beispiel werden auch Verbindungsvariablen konfiguriert.

```
ConfigureInformaticaConnectionsRedirect
 -treePath: 'ETL.INFA_SOURCE.Files'
 -connections: '{
 "ConnectionNames": [
 {
  "name": "Oracle_src",
  "newName": "postgres",
  "treePath": "Servers.ORACLE"
 }
 ]
 "ConnectionVariables": [
 {
         "name": "$Source",
         "treePath": "Servers.ORACLE"
    }
    ]
 }'
/
```

9. Konvertieren Sie Ihre Quelldatenbankschemas und Informatica ETL-Skripts.

   Das folgende Codebeispiel konvertiert alle Ihre Oracle-Quelldatenbankschemas und Ihre Informatica XML-Datei.

```
Convert
 -treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
 -treePath: 'ETL.INFA_SOURCE.Files'
/
```

10. (Optional) Speichern Sie Ihr Konvertierungsprojekt und den Bewertungsbericht. Dieser Bericht enthält die Punkte der Konversionsmaßnahmen und Empfehlungen, wie die einzelnen Punkte angegangen werden können.

    Im folgenden Codebeispiel wird Ihr Projekt gespeichert und eine Kopie des Bewertungsberichts als PDF-Datei im `C:\Temp` Ordner gespeichert.

    ```
    SaveProject
    /
    SaveReportPDF
     -treePath: 'ETL.INFA_SOURCE.Files'
     -file:'C:\Temp\Informatica.pdf'
    /
    ```

11. Speichern Sie Ihre konvertierte Informatica XML-Datei.

    Im folgenden Codebeispiel wird die konvertierte XML-Datei im `C:\Temp` Ordner gespeichert. Sie haben diesen Ordner im vorherigen Schritt mit dem `AddTarget` Befehl angegeben.

    ```
    SaveTargetInformaticaXML
    -treePath: 'ETL.INFA_TARGET.Files'
    /
    ```

12. Speichern Sie Ihr Skript als `.scts` Datei und führen Sie es mit dem `RunSCTBatch` Befehl in der AWS SCT CLI aus. Weitere Informationen finden Sie unter AWS SCT CLI-Skriptmodus.

    Im folgenden Beispiel wird das `Informatica.scts` Skript im `C:\Temp` Ordner ausgeführt. Sie können dieses Beispiel in Windows verwenden.

    ```
    RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
    ```

    Wenn Sie Ihre Informatica-ETL-Quellskripte bearbeiten, führen Sie das AWS SCT CLI-Skript erneut aus.

SSIS wird in mit konvertiert AWS GlueAWS SCT

Im Folgenden erfahren Sie, wie Sie Microsoft SQL Server Integration Services (SSIS) -Pakete in AWS Glue Using AWS SCT konvertieren.

Um Microsoft SSIS-Pakete zu konvertieren AWS Glue, stellen Sie sicher, dass Sie AWS SCT Version 1.0.642 oder höher verwenden. Sie benötigen außerdem ein SSIS-Projekt mit ETL-Paketen — `.dtsx.conmgr`, und `.params` Dateien im lokalen Ordner.

Sie benötigen keinen installierten SSIS-Server. Der Konvertierungsprozess durchläuft die lokalen SSIS-Dateien.

Um ein SSIS-Paket zu konvertieren, verwenden Sie AWS GlueAWS SCT

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen, um Ihrem Projekt ein neues SSIS-Quellpaket hinzuzufügen.
3. Wählen Sie SQL Server Integration Services und gehen Sie wie folgt vor:

   - Verbindungsname — Geben Sie den Namen für Ihre Verbindung ein. AWS SCT zeigt diesen Namen in der Metadatenstruktur an.
   - SSIS-Paketordner — Wählen Sie den Pfad zu Ihrem SSIS-Projektordner mit Paketen aus.

   AWS SCT liest die Projektdateien (Dateien mit den Erweiterungen `.dtsx .conmgr` oder`.params`) aus dem lokalen Ordner und analysiert sie. Anschließend werden sie in einem AWS SCT Kategorienbaum organisiert.
4. Wählen Sie im Menü die Option Ziel hinzufügen aus, um eine neue Zielplattform für die Konvertierung Ihrer SSIS-Quellpakete hinzuzufügen.
5. Wählen Sie Folgendes aus AWS Glueund vervollständigen Sie es:

   - Verbindungsname — Geben Sie den Namen für Ihre Verbindung ein. AWS SCT zeigt diesen Namen in der Metadatenstruktur an.
   - Aus AWS Profil kopieren — Wählen Sie das zu verwendende Profil aus.
   - AWS Zugangsschlüssel — Geben Sie Ihren AWS Zugangsschlüssel ein.
   - AWS geheimer Schlüssel — Geben Sie Ihren AWS geheimen Schlüssel ein.

- Region — Wählen Sie die Region AWS-Region , die Sie verwenden möchten, aus der Liste aus.
- Amazon S3 S3-Bucket-Ordner — Geben Sie den Ordnerpfad für den Amazon S3 S3-Bucket ein, den Sie verwenden möchten.

Sie können ein virtuelles AWS Glue Ziel verwenden. In diesem Fall müssen Sie die Verbindungsanmeldedaten nicht angeben. Weitere Informationen finden Sie unter the section called “Zuordnung virtueller Ziele”.

6. Erstellen Sie eine neue Zuordnungsregel, die Ihr SSIS-Quellpaket und Ihr AWS Glue Ziel enthält. Weitere Informationen finden Sie unter the section called “Neue Datentyp-Zuordnung”.
7. Wählen Sie im Menü Ansicht die Option Hauptansicht aus.
8. Öffnen Sie in der SSIS-Baumansicht das Kontextmenü (mit der rechten Maustaste) für Verbindungs-Manager und wählen Sie dann Verbindungen konfigurieren aus.
9. Konfigurieren Sie den Projektverbindungsmanager.

   Um eine Verbindungszuordnung für SSIS-Verbindungs-Manager zu konfigurieren, geben Sie die AWS Glue Verbindung für den entsprechenden SSIS-Verbindungs-Manager an. Stellen Sie sicher, dass Ihre AWS Glue Verbindungen bereits erstellt wurden.

   a. Wählen Sie unter Verbindungen die Option Projektverbindungen aus.
   b. Wählen Sie für die Glue-Katalogverbindung die entsprechende AWS Glue Verbindung aus.
10. Konfigurieren Sie den Paketverbindungsmanager:

    a. Wählen Sie unter Verbindungen Ihr Paket aus.
    b. Wählen Sie für die Glue-Katalogverbindung die entsprechende AWS Glue Verbindung aus.
    c. Wiederholen Sie diese Aktionen für alle Verbindungen, die für Ihr Paket verfügbar sind.
11. Wählen Sie Anwenden aus.
12. Konvertiere dein Paket. Suchen Sie in der Quellbaumansicht nach Paketen. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr Paket und wählen Sie dann Paket konvertieren.
13. Speichern Sie das konvertierte Skript in Amazon S3. Suchen Sie in der Zielstrukturansicht nach Paketskripten. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr konvertiertes Skript und wählen Sie dann In S3 speichern.

14. Konfigurieren Sie Ihren AWS Glue Job. Suchen Sie in der Zielstrukturansicht nach Paketskripten. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr konvertiertes Skript und wählen Sie dann AWS Glue Job konfigurieren.
15. Füllen Sie die drei Konfigurationsabschnitte aus:

    a. Füllen Sie den Abschnitt Datenfluss entwerfen aus:

    - Ausführungsstrategie — Wählen Sie aus, wie Ihr Job ETL-Skripts ausführen soll. Wählen Sie SEQUENTIAL, um die Skripts in der Reihenfolge auszuführen, die im Assistenten angegeben ist. Wählen Sie PARALLEL, um die Skripts parallel auszuführen und dabei die im Assistenten angegebene Reihenfolge zu ignorieren.
    - Skripte — Wählen Sie den Namen Ihres konvertierten Skripts.
    - Wählen Sie Weiter.

    b. Füllen Sie den Abschnitt Auftragseigenschaften aus:

    - Name — Geben Sie den Namen Ihres AWS Glue Jobs ein.
    - IAM-Rolle — Wählen Sie die IAM-Rolle aus, die für die Autorisierung von Ressourcen verwendet wird, die für die Ausführung des Jobs und den Zugriff auf Datenspeicher verwendet werden.
    - Name der Skriptdatei — Geben Sie den Namen Ihres konvertierten Skripts ein.
    - S3-Pfad der Skriptdatei — Geben Sie den Amazon S3 S3-Pfad zu Ihrem konvertierten Skript ein.
    - Skript mit SSE-S3 verschlüsseln — Wählen Sie diese Option, um Daten mithilfe serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) zu schützen.
    - Temporäres Verzeichnis — Geben Sie den Amazon S3 S3-Pfad zu einem temporären Verzeichnis für Zwischenergebnisse ein. AWS Glue und AWS Glue integrierte Transformationen verwenden dieses Verzeichnis, um in Amazon Redshift zu lesen oder zu schreiben.
    - AWS SCT generiert automatisch den Pfad für Python-Bibliotheken. Sie können diesen Pfad unter Generierter Python-Bibliothekspfad überprüfen. Sie können diesen automatisch generierten Pfad nicht bearbeiten. Um zusätzliche Python-Bibliotheken zu verwenden, geben Sie den Pfad im Feld Benutzer-Python-Bibliothekspfad ein.
    - Benutzer-Python-Bibliothekspfad — Geben Sie die Pfade für zusätzliche Benutzer-Python-Bibliotheken ein. Trennen Sie Amazon S3 S3-Pfade durch Kommas.

- Pfad für abhängige JAR-Dateien — Geben Sie die Pfade für abhängige JAR-Dateien ein. Trennen Sie Amazon S3 S3-Pfade durch Kommas.
- Pfad der referenzierten Dateien — Geben Sie die Pfade für zusätzliche Dateien ein, z. B. Konfigurationsdateien, die für Ihr Skript erforderlich sind. Trennen Sie Amazon S3 S3-Pfade durch Kommas.
- Maximale Kapazität — Geben Sie die maximale Anzahl von AWS Glue Datenverarbeitungseinheiten (DPUs) ein, die zugewiesen werden können, wenn dieser Job ausgeführt wird. Sie können eine Ganzzahl zwischen 2 und 100 eingeben. Der Standardwert ist 2.
- Max. Parallelität — Geben Sie die maximale Anzahl gleichzeitiger Läufe ein, die für diesen Job zulässig sind. Die Standardeinstellung ist 1. AWS Glue gibt einen Fehler zurück, wenn dieser Schwellenwert erreicht ist.
- Job-Timeout (Minuten) — Geben Sie den Timeout-Wert für Ihren ETL-Job ein, um zu verhindern, dass Jobs außer Kontrolle geraten. Der Standardwert für Batchaufträge beträgt 2880 Minuten (48 Stunden). Wenn der Job dieses Limit überschreitet, ändert sich der Status der Auftragsausführung auf. `TIMEOUT`
- Schwellenwert für Verzögerungsbenachrichtigungen (Minuten) — Geben Sie den Schwellenwert in Minuten ein, bevor eine Verzögerungsbenachrichtigung AWS SCT gesendet wird.
- Anzahl der Wiederholungen — Geben Sie ein, wie oft (0—10) der Job automatisch neu gestartet werden AWS Glue soll, falls er fehlschlägt. Jobs, die das Timeout-Limit erreichen, werden nicht neu gestartet. Der Standardwert ist 0.
- Wählen Sie Weiter.

c. Konfigurieren Sie die erforderlichen Verbindungen:

i. Wählen Sie unter Alle Verbindungen die erforderlichen AWS Glue Verbindungen aus und fügen Sie sie der Liste Ausgewählte Verbindungen hinzu.

ii. Wählen Sie Finish (Abschließen).

16. Erstellen Sie einen konfigurierten AWS Glue Job. Suchen und erweitern Sie in der Zielstrukturansicht ETL-Jobs. Öffnen Sie das Kontextmenü (Rechtsklick) für den ETL-Job, den Sie konfiguriert haben, und wählen Sie dann AWS Glue Job erstellen.
17. Führen Sie den AWS Glue Job aus:

a. Öffnen Sie die AWS Glue Konsole unter https://console.aws.amazon.com/glue/.

b. Wählen Sie im Navigationsbereich die Option Jobs (Aufträge) aus.

c. Wählen Sie Job hinzufügen und wählen Sie dann den Job aus, den Sie ausführen möchten.

d. Wählen Sie auf der Registerkarte Aktionen die Option Job ausführen aus.

SSIS-Komponenten, die konvertiert AWS SCT werden können in AWS Glue

Sie können AWS SCT damit Datenfluss- und Ablaufsteuerungskomponenten sowie Container, Parameter und Variablen konvertieren.

Zu den unterstützten Datenflusskomponenten gehören die folgenden:

- ADO NET-Ziel
- ADO NET-Quelle
- Aggregate
- Cache-Transformation
- Transformation der Zeichentabelle
- Bedingte Split-Transformation
- Transformation für Spalten kopieren
- Transformation zur Datenkonvertierung
- Transformation abgeleiteter Spalten
- Excel-Ziel
- Excel-Quelle
- Transformation für Spalten exportieren
- Flatfile-Ziel
- Flatfile-Quelle
- Transformation für Fuzzy-Suche
- Transformation für Spalten importieren
- Transformation für Suchvorgänge
- Transformation zusammenführen
- Transformation zusammenführen
- Multicast-Transformation
- ODBC-Ziel
- ODBC-Quelle

- Transformation von OLE DB-Befehlen
- OLE DB-Ziel
- OLE DB-Quelle
- Prozentuale Transformation der Stichproben
- Pivot-Transformation
- Ziel der Rohdatei
- Quelle der Rohdatei
- RecordSet Ziel
- Transformation zur Zeilenanzahl
- Transformation für Zeilenstichproben
- Transformation sortieren
- SQL Server-Ziel
- Alle Transformationen vereinigen
- Unpivo-Transformation
- XML-Quelle

Zu den unterstützten Kontrollflusskomponenten gehören:

- Masseneinfügungsaufgabe
- Aufgabe „Paket ausführen“
- Aufgabe „SQL ausführen“
- Aufgabe „T-SQL-Anweisung ausführen“
- Aufgabe „Ausdruck“
- Dateisystem-Aufgabe
- Aufgabe „Bediener benachrichtigen“
- Aufgabe „E-Mail senden“

Zu den unterstützten SSIS-Containern gehören die folgenden:

- Für Loop Container
- Für jeden Loop-Container
- Sequenz-Container

SSIS-Pakete werden in with konvertiert AWS Glue StudioAWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um Microsoft SQL Server Integration Services (SSIS) -Pakete in zu AWS Glue Studio konvertieren.

Ein SSIS-Paket enthält die erforderlichen Komponenten, z. B. den Verbindungs-Manager, Aufgaben, Ablaufsteuerung, Datenfluss, Parameter, Ereignishandler und Variablen, um eine bestimmte ETL-Task (Extrahieren, Transformieren und Laden) auszuführen. AWS SCT konvertiert SSIS-Pakete in ein Format, das kompatibel ist mit. AWS Glue Studio Nachdem Sie Ihre Quelldatenbank auf die migriert haben AWS Cloud, können Sie diese konvertierten AWS Glue Studio Jobs ausführen, um ETL-Aufgaben auszuführen.

Um Microsoft SSIS-Pakete zu konvertieren AWS Glue Studio, stellen Sie sicher, dass Sie AWS SCT Version 1.0.661 oder höher verwenden.

Themen

- Voraussetzungen
- Hinzufügen von SSIS-Paketen zu Ihrem Projekt AWS SCT
- SSIS-Pakete werden in mit konvertiert AWS Glue StudioAWS SCT
- AWS Glue Studio Jobs mit dem konvertierten Code erstellen
- Erstellen eines Bewertungsberichts für ein SSIS-Paket mit AWS SCT
- SSIS-Komponenten, die konvertiert AWS SCT werden können in AWS Glue Studio

Voraussetzungen

In diesem Abschnitt erfahren Sie mehr über die erforderlichen Aufgaben für die Konvertierung von SSIS-Paketen in. AWS Glue Zu diesen Aufgaben gehört das Erstellen der erforderlichen AWS Ressourcen in Ihrem Konto.

Sie können AWS Identity and Access Management (IAM) verwenden, um Richtlinien und Rollen zu definieren, die für den Zugriff auf Ressourcen erforderlich sind, die AWS Glue Studio verwendet werden. Weitere Informationen finden Sie unter IAM-Berechtigungen für den AWS Glue Studio Benutzer.

Nachdem Sie Ihre Quellskripte AWS SCT konvertiert haben AWS Glue Studio, laden Sie die konvertierten Skripts in einen Amazon S3 S3-Bucket hoch. Stellen Sie sicher, dass Sie diesen

Amazon S3 S3-Bucket erstellen und ihn in den AWS Serviceprofileinstellungen auswählen. Weitere Informationen zum Erstellen eines S3-Buckets finden Sie unter Erstellen Sie Ihren ersten S3-Bucket im Amazon Simple Storage Service-Benutzerhandbuch.

Um sicherzustellen, dass dieser eine Verbindung zu Ihrem Datenspeicher herstellen AWS Glue Studio kann, erstellen Sie einen benutzerdefinierten Connector und eine Verbindung. Speichern Sie Datenbankanmeldedaten außerdem in AWS Secrets Manager.

Um einen benutzerdefinierten Connector zu erstellen

1. Laden Sie den JDBC-Treiber für Ihren Datenspeicher herunter. Weitere Hinweise zu den verwendeten JDBC-Treibern finden Sie unter. AWS SCT Installation von JDBC-Treibern für AWS Schema Conversion Tool
2. Laden Sie diese Treiberdatei in Ihren Amazon S3 S3-Bucket hoch. Weitere Informationen finden Sie unter Hochladen eines Objekts in Ihren Bucket im Amazon Simple Storage Service-Benutzerhandbuch.
3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Glue Studio Konsole unter https://console.aws.amazon.com/gluestudio/.
4. Wählen Sie Connectors und anschließend Create Custom Connector aus.
5. Wählen Sie für Connector S3 URL die Option Browse S3 und wählen Sie die JDBC-Treiberdatei aus, die Sie in Ihren Amazon S3 S3-Bucket hochgeladen haben.
6. Geben Sie einen aussagekräftigen Namen für Ihren Connector ein. Geben Sie z. B. ei **`SQLServer`**.
7. Wählen Sie als Connector-Typ JDBC aus.
8. Geben Sie unter Klassenname den Namen der Hauptklasse für Ihren JDBC-Treiber ein. Geben Sie für SQL Server ein. **`com.microsoft.sqlserver.jdbc.SQLServerDriver`**
9. Geben Sie für JDBC URL base die JDBC-Basis-URL ein. Die Syntax der JDBC-Basis-URL hängt von Ihrer Quelldatenbank-Engine ab. Verwenden Sie für SQL Server das folgende Format:. **`jdbc:sqlserver://$<host>:$<port>;databaseName=$<dbname>;user=$<username>;password=$<password>`**

   Stellen Sie sicher,*`<host>`*,, *`<port>`* *`<dbname>`**`<username>`*, und durch Ihre Werte zu *`<password>`* ersetzen.
10. Geben Sie als URL-Parametertrennzeichen das Semikolon () ein. **`;`**
11. Wählen Sie Konnektor erstellen.

Um Datenbankanmeldedaten zu speichern in AWS Secrets Manager

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Secrets Manager Konsole unter https://console.aws.amazon.com/secretsmanager/.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Führen Sie auf der Seite Choose secret type (Secret-Typ auswählen) die folgenden Schritte aus:

   a. Wählen Sie unter Geheimtyp die Option Anderer Geheimtyp aus.

   b. Geben Sie für Schlüssel/Wert-Paare die folgenden Schlüssel ein:**`host`**,, **`port`** **`dbnameusername`**, und. **`password`**

      Geben Sie als Nächstes Ihre Werte für diese Schlüssel ein.
4. Geben Sie auf der Seite Geheimnis konfigurieren einen aussagekräftigen Wert für Secret-Name ein. Geben Sie z. B. ei **`SQL_Server_secret`**.
5. Wählen Sie Weiter. Wählen Sie dann auf der Seite „Rotation konfigurieren“ erneut Weiter aus.
6. Prüfen Sie auf der Seite Review (Prüfen) die Secret-Details und wählen Sie Store (Speichern).

Um eine Verbindung für Ihren Connector herzustellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Glue Studio Konsole unter https://console.aws.amazon.com/gluestudio/.
2. Wählen Sie den Connector aus, für den Sie eine Verbindung herstellen möchten, und klicken Sie dann auf Verbindung erstellen.
3. Geben Sie auf der Seite Verbindung erstellen einen aussagekräftigen Namen für Ihre Verbindung ein. Geben Sie z. B. ei **`SQL-Server-connection`**.
4. Wählen Sie für AWS Secret den Secret aus, den Sie in AWS Secrets Manager erstellt haben.
5. Konfigurieren Sie die Netzwerkoptionen und wählen Sie dann Verbindung erstellen.

Jetzt können Sie einen AWS Glue Studio Job mit einem benutzerdefinierten Connector erstellen. Weitere Informationen finden Sie unter AWS Glue Studio Jobs erstellen.

Hinzufügen von SSIS-Paketen zu Ihrem Projekt AWS SCT

Sie können einem einzelnen AWS SCT Projekt mehrere SSIS-Pakete hinzufügen.

Um Ihrem Projekt ein SSIS-Paket hinzuzufügen AWS SCT

1. Erstellen Sie ein neues Projekt mit AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü Quelle hinzufügen und wählen Sie dann SQL Server Integration Services aus.
3. Geben Sie unter Verbindungsname einen Namen für Ihre SSIS-Pakete ein. AWS SCT zeigt diesen Namen in der Struktur im linken Bereich an.
4. Geben Sie für den Ordner mit den SSIS-Paketen den Pfad zu dem Ordner mit den SSIS-Quellpaketen ein.
5. Wählen Sie im Menü die Option Ziel hinzufügen und wählen Sie dann. AWS Glue Studio

   Um eine Verbindung herzustellen AWS Glue Studio, AWS SCT verwendet Ihr AWS Profil. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.
6. Erstellen Sie eine Zuordnungsregel, die Ihr SSIS-Quellpaket und Ihr AWS Glue Studio Ziel enthält. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.
7. Stellen Sie AWS Glue Studio Verbindungen in der AWS Glue Studio Konsole her. Weitere Informationen finden Sie unter Verbindungen für Konnektoren erstellen.
8. Wählen Sie in der linken Baumstruktur Verbindungsmanager aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Verbindungen konfigurieren.

   AWS SCT zeigt das Fenster Verbindungen konfigurieren an.
9. Wählen Sie für jede SSIS-Quellverbindung eine AWS Glue Studio Verbindung aus.

SSIS-Pakete werden in mit konvertiert AWS Glue StudioAWS SCT

Im Folgenden erfahren Sie, wie Sie SSIS-Pakete in Using umwandeln. AWS Glue Studio AWS SCT

Um ein SSIS-Paket zu konvertieren AWS Glue Studio

1. Fügen Sie Ihr SSIS-Paket zu Ihrem AWS SCT Projekt hinzu. Weitere Informationen finden Sie unter Hinzufügen von SSIS-Paketen zu Ihrem Projekt AWS SCT .
2. Erweitern Sie im linken Bereich die ETL - und SSIS-Knoten.
3. Wählen Sie Pakete, öffnen Sie das Kontextmenü (mit der rechten Maustaste) und wählen Sie dann Paket konvertieren.

AWS SCT konvertiert Ihre ausgewählten SSIS-Pakete in JSON-Dateien. Diese JSON-Objekte stellen einen Knoten in einem gerichteten azyklischen Graphen (DAG) dar. Suchen Sie Ihre konvertierten Dateien im DAGs Knoten Package in der rechten Baumstruktur.

4. Wählen Sie Package DAGs, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann In Amazon S3 speichern.

   Jetzt können Sie diese Skripts verwenden, um Jobs in der zu erstellen AWS Glue Studio.

AWS Glue Studio Jobs mit dem konvertierten Code erstellen

Nachdem Sie Ihre SSIS-Quellpakete konvertiert haben, können Sie die konvertierten JSON-Dateien verwenden, um AWS Glue Studio Jobs zu erstellen.

Um einen Job zu erstellen AWS Glue Studio

1. Wählen Sie DAGs in der rechten Baumstruktur die Option Package aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann AWS Glue Studio Job konfigurieren.
2. (Optional) Wenden Sie das Erweiterungspaket an, in dem SSIS-Funktionen emuliert werden. AWS Glue Studio
3. Das Fenster AWS Glue Studio Job konfigurieren wird geöffnet.

   Füllen Sie den Abschnitt Grundlegende Jobeigenschaften aus:

   - Name — Geben Sie einen Namen für Ihren AWS Glue Studio Job ein.
   - Name der Skriptdatei — Geben Sie einen Namen für Ihr Job-Skript ein.
   - Jobparameter — Fügen Sie Parameter hinzu und geben Sie deren Werte ein.

   Wählen Sie Weiter.
4. Füllen Sie den Abschnitt „Erweiterte Auftragseigenschaften“ aus:

   - IAM-Rolle — Wählen Sie die IAM-Rolle aus, die für die Autorisierung von AWS Glue Studio und den Zugriff auf Datenspeicher verwendet wird.
   - S3-Pfad der Skriptdatei — Geben Sie den Amazon S3 S3-Pfad zu Ihrem konvertierten Skript ein.

- Temporäres Verzeichnis — Geben Sie den Amazon S3 S3-Pfad zu einem temporären Verzeichnis für Zwischenergebnisse ein. AWS Glue Studio verwendet dieses Verzeichnis, um in Amazon Redshift zu lesen oder zu schreiben.
- AWS SCT generiert automatisch den Pfad für Python-Bibliotheken. Sie können diesen Pfad unter Generierter Python-Bibliothekspfad überprüfen. Sie können diesen automatisch generierten Pfad nicht bearbeiten. Um zusätzliche Python-Bibliotheken zu verwenden, geben Sie den Pfad im Feld Benutzer-Python-Bibliothekspfad ein.
- Benutzer-Python-Bibliothekspfad — Geben Sie die Pfade für zusätzliche Benutzer-Python-Bibliotheken ein. Trennen Sie Amazon S3 S3-Pfade durch Kommas.
- Pfad für abhängige Dateien — Geben Sie die Pfade für abhängige `*.jar` Dateien ein. Trennen Sie Amazon S3 S3-Pfade durch Kommas.
- Pfad der referenzierten Dateien — Geben Sie die Pfade für zusätzliche Dateien ein, z. B. Konfigurationsdateien, die für Ihr Skript erforderlich sind. Trennen Sie Amazon S3 S3-Pfade durch Kommas.
- Arbeitstyp — Wählen Sie `G.1X` oder`G.2X`.

  Wenn Sie wählen, wird `G.1X` jeder Worker einer DPU (4 vCPU, 16 GB Arbeitsspeicher und 64 GB Festplatte) zugeordnet.

  Wenn Sie wählen, wird `G.2X` jeder Worker 2 DPU (8 vCPU, 32 GB Arbeitsspeicher und 128 GB Festplatte) zugeordnet.
- Angeforderte Anzahl von Workern — Geben Sie die Anzahl der Worker ein, die bei der Ausführung des Jobs zugewiesen werden.
- Max. Parallelität — Geben Sie die maximale Anzahl gleichzeitiger Läufe ein, die für diesen Job zulässig sind. Die Standardeinstellung ist 1. AWS Glue gibt einen Fehler zurück, wenn dieser Schwellenwert erreicht ist.
- Job-Timeout (Minuten) — Geben Sie den Timeout-Wert für Ihren ETL-Job ein, um zu verhindern, dass Jobs außer Kontrolle geraten. Die Standardeinstellung für Batch-Jobs ist 2.880 Minuten (48 Stunden). Wenn der Job dieses Limit überschreitet, ändert sich der Status der Auftragsausführung auf`TIMEOUT`.
- Schwellenwert für Verzögerungsbenachrichtigungen (Minuten) — Geben Sie den Schwellenwert in Minuten ein, bevor eine Verzögerungsbenachrichtigung AWS SCT gesendet wird.

- Anzahl der Wiederholungen — Geben Sie ein, wie oft (0—10) der Job automatisch neu gestartet werden AWS Glue soll, falls er fehlschlägt. Jobs, die das Timeout-Limit erreichen, werden nicht neu gestartet. Der Standardwert ist 0.

Wählen Sie Finish (Abschließen).

AWS SCT konfiguriert Ihre ausgewählten Jobs. AWS Glue Studio

5. Suchen Sie Ihre konfigurierten Jobs unter ETL-Jobs in der rechten Baumstruktur. Wählen Sie Ihren konfigurierten Job aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann AWS Glue Studio Job erstellen.
6. Wählen Sie „Status anwenden“ und stellen Sie sicher, dass der Statuswert für Ihren Job „Erfolgreich“ lautet.
7. Öffnen Sie die AWS Glue Studio Konsole, wählen Sie „Aktualisieren“ und wählen Sie Ihren Job aus. Wählen Sie anschließend Run (Ausführen) aus.

Erstellen eines Bewertungsberichts für ein SSIS-Paket mit AWS SCT

Der Bewertungsbericht zur ETL-Migration enthält Informationen zur Konvertierung Ihrer SSIS-Pakete in ein Format, das mit kompatibel ist. AWS Glue Studio Der Bewertungsbericht enthält Aktionspunkte für die Komponenten Ihrer SSIS-Pakete. Diese Aktionspunkte zeigen, welche Komponenten nicht automatisch konvertiert werden AWS SCT können.

Um einen Bewertungsbericht zur ETL-Migration zu erstellen

1. Erweitern Sie den SSIS-Knoten unter ETL im linken Bereich.
2. Wählen Sie Pakete, öffnen Sie das Kontextmenü (mit der rechten Maustaste) und wählen Sie dann Bericht erstellen aus.
3. Rufen Sie die Registerkarte „Zusammenfassung“ auf. Hier AWS SCT werden die zusammenfassenden Informationen aus dem ETL-Migrationsbewertungsbericht angezeigt. Es enthält Konvertierungsergebnisse für alle Komponenten Ihrer SSIS-Pakete.
4. (Optional) Speichern Sie eine lokale Kopie des ETL-Migrationsbewertungsberichts entweder als PDF-Datei oder als CSV-Datei (Comma-Separated Values):

   - Um den ETL-Migrationsbewertungsbericht als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um den Bewertungsbericht zur ETL-Migration als CSV-Datei zu speichern, wählen Sie oben rechts die Option In CSV speichern aus.

  AWS SCT erstellt drei CSV-Dateien. Diese Dateien enthalten Aktionspunkte, empfohlene Aktionen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.

5. Wählen Sie die Registerkarte Aktionspunkte. Diese Registerkarte enthält eine Liste von Elementen, in die eine manuelle Konvertierung erforderlich ist AWS Glue Studio. Wenn Sie einen Aktionspunkt aus der Liste auswählen, wird der Punkt aus Ihrem SSIS-Quellpaket AWS SCT hervorgehoben, für den der Aktionspunkt gilt.

SSIS-Komponenten, die konvertiert AWS SCT werden können in AWS Glue Studio

Sie können AWS SCT es verwenden, um SSIS-Datenflusskomponenten und -parameter in zu konvertieren. AWS Glue Studio

Zu den unterstützten Datenflusskomponenten gehören:

- ADO NET-Ziel
- ADO NET-Quelle
- Aggregate
- Zeichentabelle
- Bedingte Trennung
- Spalte kopieren
- Datenkonvertierung
- Abgeleitete Spalte
- Nachschlagen
- Merge
- Merge Join
- Multicast

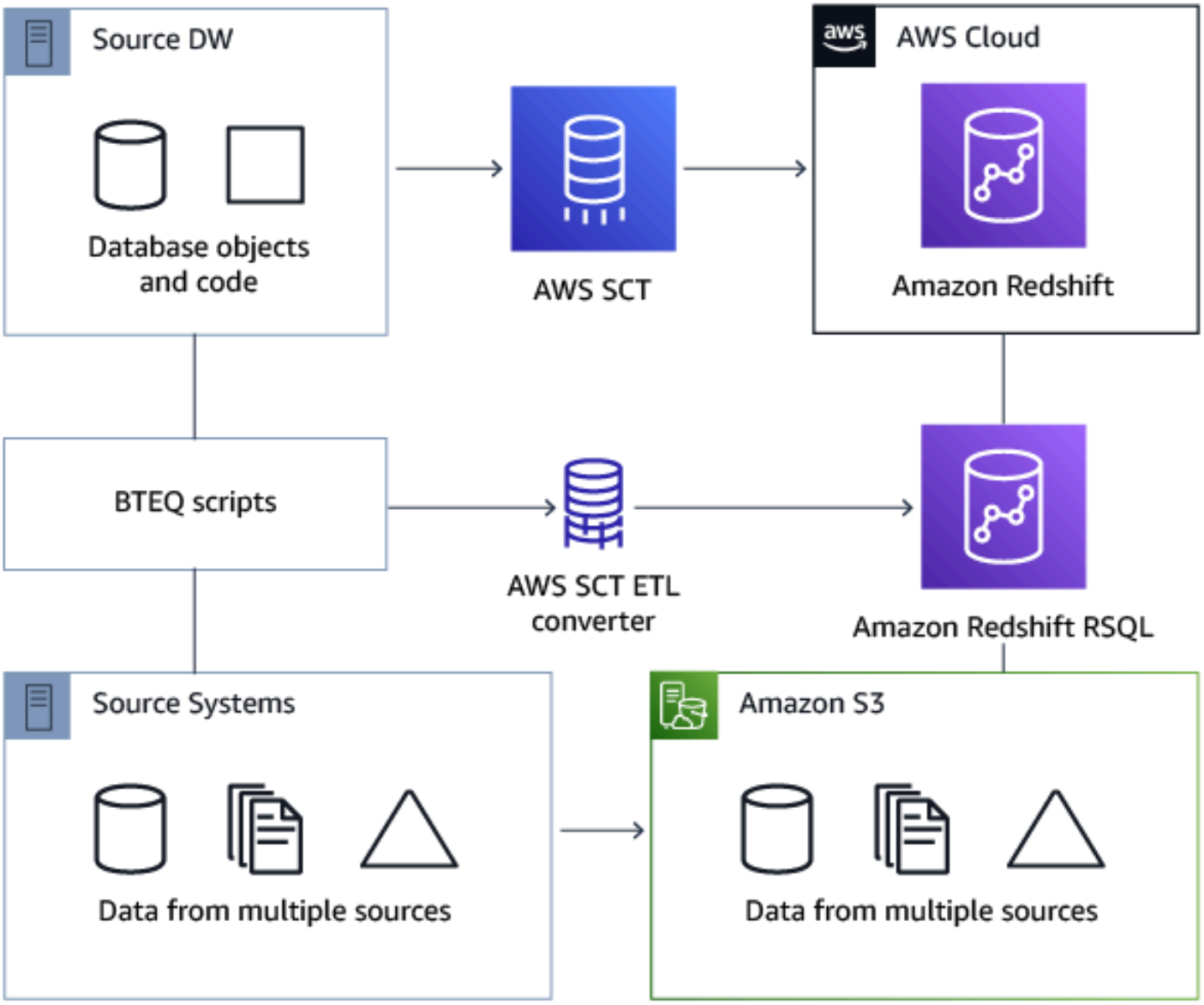
- ODBCDestination
- ODBCSource
- OLEDBDestination
- OLEDBSource
- Anzahl der Zeilen
- Sortierung
- SQL Server-Ziel
- Alle vereinigen

AWS SCT kann mehr SSIS-Komponenten in konvertieren. AWS Glue Weitere Informationen finden Sie unter SSIS-Komponenten, die konvertiert AWS SCT werden können in AWS Glue.

Konvertieren von Teradata BTEQ-Skripten nach Amazon Redshift RSQL mit AWS SCT

Sie können die AWS Schema Conversion Tool (AWS SCT) verwenden, um Teradata Basic Teradata Query (BTEQ) -Skripts in Amazon Redshift RSQL zu konvertieren.

Das folgende Architekturdiagramm zeigt das Datenbankmigrationsprojekt, das die Konvertierung von ETL-Skripts (Extrahieren, Transformieren und Laden) nach Amazon Redshift RSQL umfasst.



Themen

- Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT
- Konfiguration von Substitutionsvariablen in BTEQ-Skripten mit AWS SCT
- Konvertieren von Teradata BTEQ-Skripten nach Amazon Redshift RSQL mit AWS SCT
- Verwaltung von BTEQ-Skripten mit AWS SCT
- Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung mit AWS SCT
- Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte mit AWS SCT

Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT

Sie können einem einzelnen AWS SCT Projekt mehrere Skripte hinzufügen.

Um Ihrem Projekt ein BTEQ-Skript hinzuzufügen AWS SCT

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen und wählen Sie dann Teradata, um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter Teradata-Datenbanken.
3. Wählen Sie im Menü die Option Ziel hinzufügen, um Ihrem AWS SCT Projekt eine Amazon Redshift Redshift-Zieldatenbank hinzuzufügen.

   Sie können eine virtuelle Amazon Redshift Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift Redshift-Ziel umfasst. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
5. Wählen Sie im Menü Ansicht die Option Hauptansicht.
6. Erweitern Sie im linken Bereich den Knoten Scripts.
7. Wählen Sie BTEQ-Skripte, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skripten laden aus.
8. Geben Sie den Speicherort des Quellcodes für Ihre Teradata BTEQ-Skripte ein und wählen Sie Ordner auswählen.

   AWS SCT zeigt das Fenster Skripten laden an.
9. Führen Sie eine der folgenden Aktionen aus:

   a. Wenn Ihre Teradata BTEQ-Skripts die Substitutionsvariablen nicht enthalten, wählen Sie Keine Substitutionsvariablen und dann OK, um Ihrem Projekt Skripten hinzuzufügen. AWS SCT

   b. Wenn Ihre Teradata BTEQ-Skripten die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in BTEQ-Skripten.

Konfiguration von Substitutionsvariablen in BTEQ-Skripten mit AWS SCT

Ihre Teradata BTEQ-Skripts können Substitutionsvariablen enthalten. Sie können beispielsweise ein BTEQ-Skript mit Substitutionsvariablen verwenden, um denselben Befehlssatz in mehreren Datenbankumgebungen auszuführen. Sie können es verwenden AWS SCT , um Substitutionsvariablen in Ihren BTEQ-Skripten zu konfigurieren.

Bevor Sie ein BTEQ-Skript mit Substitutionsvariablen ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript UC4 (Automic) usw. verwenden. AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihnen Werte zugewiesen haben.

Um Substitutionsvariablen in Ihrem BTEQ-Skript zu konfigurieren

1. Fügen Sie Ihre BTEQ-Skripte zu Ihrem Projekt hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT .

   Wenn Sie Ihre Skripte hinzufügen, wählen Sie Substitutionsvariablen werden verwendet.
2. Geben Sie unter Variablenformat definieren einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

   Wenn die Namen Ihrer Substitutionsvariablen beispielsweise mit beginnen `${` und mit enden`}`, verwenden Sie den `\$\{\w+\}` regulären Ausdruck. Verwenden Sie den `\$\w+|\%\w+` regulären Ausdruck, um nach Substitutionsvariablen zu suchen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen.

   Reguläre Ausdrücke, die der Java-Syntax für reguläre Ausdrücke AWS SCT entsprechen. Weitere Informationen finden Sie unter java.util.regex Class Pattern in der Java-Dokumentation.
3. Wählen Sie OK, um Skripten in Ihr AWS SCT Projekt zu laden, und klicken Sie dann auf OK, um das Fenster Skripten laden zu schließen.
4. Wählen Sie Variablen, um alle erkannten Substitutionsvariablen und ihre Werte anzuzeigen.
5. Geben Sie unter Wert den Wert für die Substitutionsvariable ein.

Konvertieren von Teradata BTEQ-Skripten nach Amazon Redshift RSQL mit AWS SCT

Im Folgenden erfahren Sie, wie Sie BTEQ-ETL-Skripts mithilfe von Amazon Redshift RSQL konvertieren. AWS SCT

So konvertieren Sie ein Teradata BTEQ-Skript in Amazon Redshift RSQL

1. Fügen Sie Ihre BTEQ-Skripte zu Ihrem Projekt hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT .
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in BTEQ-Skripten.
3. Erweitern Sie im linken Bereich den Knoten Scripts.
4. Führen Sie eine der folgenden Aktionen aus:

   - Um ein einzelnes BTEQ-Skript zu konvertieren, erweitern Sie den Knoten BTEQ-Skripte, wählen Sie das zu konvertierende Skript aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option In RSQL konvertieren aus.
   - Um mehrere Skripten zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripten auswählen. Wählen Sie dann BTEQ-Skripten, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann unter Skript konvertieren die Option In RSQL konvertieren aus.

   AWS SCT konvertiert alle Ihre ausgewählten Teradata BTEQ-Skripts in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Suchen Sie Ihre konvertierten Skripten im Knoten „Skripte“ im Bereich „Zieldatenbank“.
5. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte.

Verwaltung von BTEQ-Skripten mit AWS SCT

Sie können mehrere BTEQ-Skripte hinzufügen oder ein BTEQ-Skript aus Ihrem Projekt entfernen. AWS SCT

Um Ihrem Projekt ein zusätzliches BTEQ-Skript hinzuzufügen AWS SCT

1. Erweitern Sie den Knoten Scripts im linken Bereich.
2. Wählen Sie den Knoten BTEQ-Skripte und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripten laden.
4. Geben Sie die Informationen ein, die zum Hinzufügen eines neuen BTEQ-Skripts und zum Konfigurieren von Substitutionsvariablen erforderlich sind. Weitere Informationen erhalten Sie unter Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT und Konfiguration von Substitutionsvariablen in BTEQ-Skripten.

Um ein BTEQ-Skript aus Ihrem Projekt zu entfernen AWS SCT

1. Erweitern Sie den Knoten BTEQ-Skripte unter Skripten im linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung mit AWS SCT

Ein Bewertungsbericht zur BTEQ-Skriptkonvertierung enthält Informationen zur Konvertierung der BTEQ-Befehle und SQL-Anweisungen aus Ihren BTEQ-Skripten in ein mit Amazon Redshift RSQL kompatibles Format. Der Bewertungsbericht enthält Aktionspunkte für BTEQ-Befehle und SQL-Anweisungen, die nicht konvertiert werden können. AWS SCT

Um einen Bewertungsbericht zur BTEQ-Skriptkonvertierung zu erstellen

1. Erweitern Sie im linken Bereich den Knoten BTEQ-Skripte unter Skripten.
2. Wählen Sie das zu konvertierende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie unter Bericht erstellen die Option Konvertierung zu RSQL aus.
4. Rufen Sie die Registerkarte „Zusammenfassung“ auf. Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Bewertungsbericht für das BTEQ-Skript angezeigt. Sie enthält Konvertierungsergebnisse für alle BTEQ-Befehle und SQL-Anweisungen aus Ihren BTEQ-Skripten.
5. (Optional) Speichern Sie eine lokale Kopie des Bewertungsberichts zur BTEQ-Skriptkonvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Um den Bewertungsbericht zur BTEQ-Skriptkonvertierung als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

  Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.
- Um den Bewertungsbericht zur BTEQ-Skriptkonvertierung als CSV-Datei zu speichern, wählen Sie oben rechts die Option In CSV speichern aus.

  Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.

6. Wählen Sie die Registerkarte Aktionspunkte aus. Diese Registerkarte enthält eine Liste von Elementen, die manuell in Amazon Redshift RSQL konvertiert werden müssen. Wenn Sie ein Aktionselement aus der Liste auswählen, wird das Element aus Ihrem BTEQ-Quellskript AWS SCT hervorgehoben, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT Projekts bearbeiten. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie den Knoten RSQL-Skripten unter Skripten im Bereich Zieldatenbank.
2. Wählen Sie das konvertierte Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie Speichern.

   AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Konvertierung von Shell-Skripten mit eingebetteten BTEQ-Befehlen nach Amazon Redshift RSQL mit AWS Schema Conversion Tool

Sie können die AWS Schema Conversion Tool (AWS SCT) verwenden, um Shell-Skripte mit eingebetteten Teradata Basic Teradata Query (BTEQ) -Befehlen in Shell-Skripts mit eingebetteten Amazon Redshift RSQL-Befehlen zu konvertieren.

AWS SCT extrahiert Teradata BTEQ-Befehle aus Ihren Shell-Skripten und konvertiert sie in ein mit Amazon Redshift kompatibles Format. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Ihre neue Amazon Redshift Redshift-Datenbank zu verwalten.

Sie können es auch verwenden AWS SCT , um Dateien mit Teradata BTEQ ETL-Skripts in Amazon Redshift RSQL zu konvertieren. Weitere Informationen finden Sie unter Konvertieren von Teradata BTEQ-Skripten nach Amazon Redshift RSQL mit AWS SCT.

Themen

- Hinzufügen von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu Ihrem Projekt AWS SCT
- Konfiguration von Substitutionsvariablen in Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT
- Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT
- Verwaltung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT
- Erstellen Sie einen Bewertungsbericht für eine Shell-Skriptkonvertierung mit AWS SCT
- Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte mit AWS SCT

Hinzufügen von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu Ihrem Projekt AWS SCT

Sie können einem einzelnen Projekt mehrere Skripts hinzufügen. AWS SCT

Um Ihrem AWS SCT Projekt ein Shell-Skript hinzuzufügen

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen und wählen Sie dann Teradata, um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter Teradata-Datenbanken.
3. Wählen Sie im Menü die Option Ziel hinzufügen aus, um Ihrem AWS SCT Projekt eine Amazon Redshift Redshift-Zieldatenbank hinzuzufügen.

   Sie können eine virtuelle Amazon Redshift Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift Redshift-Ziel umfasst. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
5. Wählen Sie im Menü Ansicht die Option Hauptansicht.
6. Erweitern Sie im linken Bereich den Knoten Scripts.
7. Wählen Sie Shell, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skripten laden.
8. Geben Sie den Speicherort Ihrer Quell-Shell-Skripte mit eingebetteten Teradata BTEQ-Befehlen ein und wählen Sie Ordner auswählen.

   AWS SCT zeigt das Fenster Skripten laden an.
9. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Ihre Shell-Skripten die Substitutionsvariablen nicht enthalten, wählen Sie Keine Ersatzvariablen und dann OK, um Ihrem AWS SCT Projekt Skripten hinzuzufügen.
   - Wenn Ihre Shell-Skripten die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Shell-Skripten.

Konfiguration von Substitutionsvariablen in Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Ihre Shell-Skripts können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Ersatzvariablen verwenden, um Datenbanken in verschiedenen Umgebungen zu verwalten. Sie können es verwenden AWS SCT , um Substitutionsvariablen in Ihren Shell-Skripten zu konfigurieren.

Bevor Sie BTEQ-Befehle mit Substitutionsvariablen aus einem Shell-Skript ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen in diesem Shell-Skript zuweisen. AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihnen Werte zugewiesen haben.

Um Substitutionsvariablen in Ihrem Shell-Skript zu konfigurieren

1. Fügen Sie Ihre Quell-Shell-Skripte zu Ihrem AWS SCT Projekt hinzu. Weitere Informationen finden Sie unter Hinzufügen von Shell-Skripten zu Ihrem Projekt AWS SCT .

   Wenn Sie Ihre Skripts hinzufügen, wählen Sie Substitutionsvariablen werden verwendet.

2. Geben Sie unter Variablenformat definieren einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

   Wenn die Namen Ihrer Substitutionsvariablen beispielsweise mit beginnen `${` und mit enden`}`, verwenden Sie den `\$\{\w+\}` regulären Ausdruck. Verwenden Sie den `\$\w+|\%\w+` regulären Ausdruck, um nach Substitutionsvariablen zu suchen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen.

   Reguläre Ausdrücke, die der Java-Syntax für reguläre Ausdrücke AWS SCT entsprechen. Weitere Informationen finden Sie unter java.util.regex Class Pattern in der Java-Dokumentation.
3. Wählen Sie OK, um Skripten in Ihr AWS SCT Projekt zu laden, und klicken Sie dann auf OK, um das Fenster Skripten laden zu schließen.
4. Wählen Sie Variablen, um alle erkannten Substitutionsvariablen und ihre Werte anzuzeigen.
5. Geben Sie unter Wert den Wert für die Substitutionsvariable ein.

Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Im Folgenden erfahren Sie, wie Sie Shell-Skripte mit eingebetteten Teradata BTEQ-Befehlen in Shell-Skripte mit eingebetteten Amazon Redshift RSQL-Befehlen konvertieren können. AWS SCT

Um ein Shell-Skript zu konvertieren

1. Fügen Sie Ihre Shell-Skripte zu Ihrem AWS SCT Projekt hinzu. Weitere Informationen finden Sie unter Hinzufügen von Shell-Skripten zu Ihrem Projekt AWS SCT .
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Shell-Skripten.
3. Erweitern Sie im linken Bereich den Knoten Scripts.
4. Führen Sie eine der folgenden Aktionen aus:

   - Um BTEQ-Befehle aus einem einzelnen Shell-Skript zu konvertieren, erweitern Sie den Shell-Knoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Skript konvertieren.
   - Um mehrere Skripten zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripten auswählen. Wählen Sie dann Shell, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skript konvertieren.

5. Wählen Sie OK aus.

   AWS SCT konvertiert BTEQ-Befehle in Ihren ausgewählten Shell-Skripten in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Suchen Sie Ihre konvertierten Skripten im Knoten „Skripte“ im Bereich „Zieldatenbank“.

6. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte.

Verwaltung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Sie können mehrere Shell-Skripte hinzufügen oder ein Shell-Skript aus Ihrem Projekt entfernen. AWS SCT

Um Ihrem AWS SCT Projekt ein neues Shell-Skript hinzuzufügen

1. Erweitern Sie den Knoten Scripts im linken Bereich.
2. Wählen Sie den Shell-Knoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripten laden.
4. Geben Sie die Informationen ein, die zum Hinzufügen eines neuen Shell-Skripts und zum Konfigurieren von Substitutionsvariablen erforderlich sind. Weitere Informationen erhalten Sie unter Hinzufügen von Shell-Skripten zu Ihrem Projekt AWS SCT und Konfiguration von Substitutionsvariablen in Shell-Skripten.

Um ein Shell-Skript aus Ihrem Projekt zu AWS SCT entfernen

1. Erweitern Sie den Shell-Knoten unter Skripten im linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen Sie einen Bewertungsbericht für eine Shell-Skriptkonvertierung mit AWS SCT

Der Bewertungsbericht zur Shell-Skriptkonvertierung enthält Informationen zur Konvertierung der BTEQ-Befehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten in ein mit Amazon Redshift RSQL kompatibles Format. Der Bewertungsbericht enthält Aktionspunkte für BTEQ-Befehle und SQL-Anweisungen, die AWS SCT nicht konvertiert werden können.

Um einen Bewertungsbericht zur Konvertierung von Shell-Skripten zu erstellen

1. Erweitern Sie den Shell-Knoten unter Skripten im linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Bericht erstellen.
3. Rufen Sie die Registerkarte „Zusammenfassung“ auf. Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Shell-Skriptbewertungsbericht angezeigt. Es enthält Konvertierungsergebnisse für alle BTEQ-Befehle und SQL-Anweisungen aus Ihren Quellskripten.
4. (Optional) Speichern Sie eine lokale Kopie des Bewertungsberichts zur Shell-Skriptkonvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Um den Bewertungsbericht zur Shell-Skriptkonvertierung als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.
   - Um den Bewertungsbericht zur Shell-Skriptkonvertierung als CSV-Datei zu speichern, wählen Sie oben rechts die Option In CSV speichern aus.

     Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.
5. Wählen Sie die Registerkarte Aktionspunkte aus. Diese Registerkarte enthält eine Liste von Elementen, die manuell in Amazon Redshift RSQL konvertiert werden müssen. Wenn Sie ein Aktionselement aus der Liste auswählen, wird das Element aus Ihrem Quell-Shell-Skript AWS SCT hervorgehoben, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT Projekts bearbeiten. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie den Knoten RSQL-Skripten unter Skripten im Bereich Zieldatenbank.
2. Wählen Sie das konvertierte Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie Speichern.

   AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Konvertieren von FastExport Skripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool

Sie können die AWS Schema Conversion Tool (AWS SCT) verwenden, um FastExport Teradata-Jobskripte in Amazon Redshift RSQL zu konvertieren.

Ein FastExport Jobskript besteht aus einer Reihe von FastExport Befehlen und SQL-Anweisungen, die Daten aus der Teradata-Datenbank auswählen und exportieren. AWS SCT konvertiert FastExport Befehle und SQL-Anweisungen in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Daten aus der Amazon Redshift Redshift-Datenbank zu exportieren.

Themen

- Hinzufügen von FastExport Jobskripten zu Ihrem Projekt AWS SCT
- Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit FastExport AWS SCT
- FastExport Teradata-Jobskripte konvertieren mit AWS SCT
- Verwaltung von FastExport Teradata-Jobskripten mit AWS SCT
- Erstellen Sie einen Bewertungsbericht für die Konvertierung eines FastExport Teradata-Jobskripts mit AWS SCT
- Bearbeiten und Speichern Ihrer konvertierten FastExport Teradata-Jobskripte mit AWS SCT

Hinzufügen von FastExport Jobskripten zu Ihrem Projekt AWS SCT

Sie können einem einzelnen AWS SCT Projekt mehrere Skripte hinzufügen.

Um Ihrem AWS SCT Projekt ein FastExport Jobskript hinzuzufügen

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen und wählen Sie dann Teradata, um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter Teradata-Datenbanken.
3. Wählen Sie im Menü die Option Ziel hinzufügen aus, um Ihrem AWS SCT Projekt eine Amazon Redshift Redshift-Zieldatenbank hinzuzufügen.

   Sie können eine virtuelle Amazon Redshift Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift Redshift-Ziel umfasst. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
5. Wählen Sie im Menü Ansicht die Option Hauptansicht.
6. Erweitern Sie im linken Bereich den Knoten Scripts.
7. Wählen Sie FastExport, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skripten laden.
8. Geben Sie den Speicherort des Quellcodes für Ihre FastExport Teradata-Jobskripte ein und wählen Sie Ordner auswählen.

   AWS SCT zeigt das Fenster Skripten laden an.
9. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Ihre FastExport Teradata-Jobskripte die Substitutionsvariablen nicht enthalten, wählen Sie Keine Substitutionsvariablen und dann OK, um Ihrem Projekt Skripten hinzuzufügen. AWS SCT
   - Wenn Ihre FastExport Teradata-Jobskripte die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten FastExport .

Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit FastExport AWS SCT

Ihre FastExport Teradata-Jobskripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten aus mehreren Datenbanken zu exportieren. Sie können es verwenden AWS SCT , um Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Bevor Sie ein FastExport Jobskript mit Substitutionsvariablen ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript UC4 (Automic) usw. verwenden. AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihnen Werte zugewiesen haben.

Um Substitutionsvariablen in Ihrem FastExport Jobskript zu konfigurieren

1. Fügen Sie Ihrem Projekt Ihre FastExport Teradata-Quelljobskripte hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von BTEQ-Skripten zu Ihrem Projekt AWS SCT .

   Wenn Sie Ihre Skripte hinzufügen, wählen Sie Substitutionsvariablen werden verwendet.
2. Geben Sie unter Variablenformat definieren einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

   Wenn die Namen Ihrer Substitutionsvariablen beispielsweise mit beginnen `${` und mit enden`}`, verwenden Sie den `\$\{\w+\}` regulären Ausdruck. Verwenden Sie den `\$\w+|\%\w+` regulären Ausdruck, um nach Substitutionsvariablen zu suchen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen.

   Reguläre Ausdrücke, die der Java-Syntax für reguläre Ausdrücke AWS SCT entsprechen. Weitere Informationen finden Sie unter java.util.regex Class Pattern in der Java-Dokumentation.
3. Wählen Sie OK, um Skripten in Ihr AWS SCT Projekt zu laden, und klicken Sie dann auf OK, um das Fenster Skripten laden zu schließen.
4. Erweitern Sie im linken Bereich den Knoten Scripts. Wählen Sie FastExportund wählen Sie dann Ihren Ordner mit Skripten aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann unter Substitutionsvariablen die Option Variablen exportieren.
5. Exportieren Sie Substitutionsvariablen für ein Skript. Erweitern Sie Ihren Ordner mit Skripten, wählen Sie Ihr Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie unter Substitutionsvariablen die Option Variablen exportieren.

6. Geben Sie den Namen der Datei mit kommagetrennten Werten (CSV) ein, in der die Substitutionsvariablen gespeichert werden sollen, und wählen Sie Speichern.
7. Öffnen Sie diese CSV-Datei und geben Sie die Werte für die Substitutionsvariablen ein.

   AWS SCT Verwendet je nach Betriebssystem unterschiedliche Formate für CSV-Dateien. Die Werte in der Datei können entweder in Anführungszeichen eingeschlossen sein oder nicht. Stellen Sie sicher, dass Sie für die Werte der Substitutionsvariablen dasselbe Format wie für die anderen Werte in der Datei verwenden. AWS SCT kann die CSV-Datei mit Werten in verschiedenen Formaten nicht importieren.
8. Speichern Sie die CSV-Datei.
9. Erweitern Sie im linken Bereich den Knoten Scripts. Wählen Sie FastExportund wählen Sie dann Ihr Skript aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann unter Substitutionsvariablen die Option Variablen importieren.
10. Wählen Sie Ihre CSV-Datei und dann Öffnen aus.
11. Wählen Sie Variablen, um alle erkannten Substitutionsvariablen und ihre Werte anzuzeigen.

FastExport Teradata-Jobskripte konvertieren mit AWS SCT

Im Folgenden erfahren Sie, wie Sie FastExport Teradata-Jobs mithilfe von Amazon Redshift RSQL konvertieren. AWS SCT

So konvertieren Sie ein FastExport Teradata-Jobskript in Amazon Redshift RSQL

1. Fügen Sie Ihre FastExport Jobskripte zu Ihrem Projekt hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von FastExport Jobskripten zu Ihrem AWS SCT Projekt.
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten FastExport .
3. Erweitern Sie im linken Bereich den Knoten Scripts.
4. Führen Sie eine der folgenden Aktionen aus:

   - Um ein einzelnes FastExport Job-Skript zu konvertieren, erweitern Sie den FastExportKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Skript konvertieren.
   - Um mehrere Skripten zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripten auswählen. Wählen Sie dann FastExport, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skript konvertieren.

AWS SCT konvertiert alle Ihre ausgewählten FastExport Teradata-Jobskripte in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Suchen Sie Ihre konvertierten Skripten im Knoten „Skripte“ im Bereich „Zieldatenbank“.

5. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihrer konvertierten FastExport Jobskripte.

Verwaltung von FastExport Teradata-Jobskripten mit AWS SCT

Sie können mehrere FastExport Teradata-Jobskripte hinzufügen oder ein FastExport Jobskript aus Ihrem Projekt entfernen. AWS SCT

Um Ihrem Projekt ein neues FastExport Jobskript hinzuzufügen AWS SCT

1. Erweitern Sie den Knoten Skripte im linken Bereich.
2. Wählen Sie den FastExportKnoten aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripten laden.
4. Geben Sie die Informationen ein, die zum Hinzufügen eines neuen FastExport Job-Skripts und zum Konfigurieren von Substitutionsvariablen erforderlich sind. Weitere Informationen erhalten Sie unter Hinzufügen von FastExport Jobskripten zu Ihrem AWS SCT Projekt und Konfiguration von Substitutionsvariablen in Jobskripten FastExport .

Um ein FastExport Jobskript aus Ihrem Projekt zu AWS SCT entfernen

1. Erweitern Sie den FastExportKnoten unter Skripte im linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen Sie einen Bewertungsbericht für die Konvertierung eines FastExport Teradata-Jobskripts mit AWS SCT

Der Bewertungsbericht zur FastExport Jobskriptkonvertierung enthält Informationen zur Konvertierung der FastExport Befehle und SQL-Anweisungen aus Ihren FastExport Skripten in ein

mit Amazon Redshift RSQL kompatibles Format. Der Bewertungsbericht enthält Aktionspunkte für FastExport Befehle und SQL-Anweisungen, die nicht konvertiert AWS SCT werden können.

Um einen Bewertungsbericht zur Skriptkonvertierung für einen FastExport Teradata-Job zu erstellen

1. Erweitern Sie den FastExportKnoten unter Skripten im linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Bericht erstellen.
3. Rufen Sie die Registerkarte „Zusammenfassung“ auf. Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Bewertungsbericht für das FastExport Jobskript angezeigt. Sie enthält Konvertierungsergebnisse für alle FastExport Befehle und SQL-Anweisungen aus Ihren Quellskripten.
4. Sie können eine lokale Kopie des Bewertungsberichts zur Konvertierung von FastExport Jobskripten entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV) speichern.

   a. Um den Bewertungsbericht zur Konvertierung von FastExport Jobskripten als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

      Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

   b. Um den Bewertungsbericht zur Konvertierung von FastExport Jobskripten als CSV-Datei zu speichern, wählen Sie oben rechts die Option In CSV speichern.

      Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.
5. Wählen Sie die Registerkarte Aktionspunkte aus. Diese Registerkarte enthält eine Liste von Elementen, die manuell in Amazon Redshift RSQL konvertiert werden müssen. Wenn Sie ein Aktionselement aus der Liste auswählen, wird das Element aus Ihrem FastExport Quell-Jobskript AWS SCT hervorgehoben, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten FastExport Teradata-Jobskripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT Projekts bearbeiten. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie den Knoten RSQL-Skripten unter Skripten im Bereich Zieldatenbank.
2. Wählen Sie das konvertierte Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie Speichern.

   AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Konvertieren von FastLoad Jobskripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool

Sie können die AWS Schema Conversion Tool (AWS SCT) verwenden, um FastLoad Teradata-Jobskripte in Amazon Redshift RSQL zu konvertieren.

Ein FastLoad Teradata-Skript besteht aus einer Reihe von Befehlen, die mehrere Sitzungen verwenden, um Daten in eine leere Tabelle in einer Teradata-Datenbank zu laden. Teradata FastLoad verarbeitet eine Reihe von Teradata-Befehlen und SQL-Anweisungen. FastLoad Die FastLoad Teradata-Befehle ermöglichen die Sitzungssteuerung und die Datenverarbeitung der Datenübertragungen. Die SQL-Anweisungen erstellen, verwalten und löschen Tabellen.

AWS SCT konvertiert FastLoad Teradata-Befehle und SQL-Anweisungen in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Daten in Ihre Amazon Redshift Redshift-Datenbank zu laden.

Themen

- Hinzufügen von FastLoad Jobskripten zu Ihrem Projekt AWS SCT
- Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit FastLoad AWS SCT
- FastLoad Teradata-Jobskripte konvertieren mit AWS SCT
- Verwaltung von FastLoad Teradata-Jobskripten mit AWS SCT
- Erstellen Sie einen Bewertungsbericht für die Konvertierung eines FastLoad Teradata-Jobskripts mit AWS SCT
- Bearbeiten und Speichern Ihrer konvertierten FastLoad Teradata-Jobskripte mit AWS SCT

Hinzufügen von FastLoad Jobskripten zu Ihrem Projekt AWS SCT

Sie können einem einzelnen AWS SCT Projekt mehrere Skripte hinzufügen.

Um Ihrem AWS SCT Projekt ein FastLoad Jobskript hinzuzufügen

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen und wählen Sie dann Teradata, um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter Teradata-Datenbanken.
3. Wählen Sie im Menü die Option Ziel hinzufügen und fügen Sie Ihrem AWS SCT Projekt eine Amazon Redshift Redshift-Zieldatenbank hinzu.

   Sie können eine virtuelle Amazon Redshift Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift Redshift-Ziel umfasst. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
5. Wählen Sie im Menü Ansicht die Option Hauptansicht.
6. Erweitern Sie im linken Bereich den Knoten Scripts.
7. Wählen Sie FastLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skripten laden.
8. Geben Sie den Speicherort Ihrer FastLoad Teradata-Quelljobskripte ein und wählen Sie Ordner auswählen aus.

   AWS SCT zeigt das Fenster Skripten laden an.
9. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Ihre FastLoad Teradata-Jobskripte die Substitutionsvariablen nicht enthalten, wählen Sie Keine Substitutionsvariablen und dann OK, um Ihrem Projekt Skripten hinzuzufügen. AWS SCT
   - Wenn Ihre FastLoad Teradata-Jobskripte die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten FastLoad .

Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit FastLoad AWS SCT

Ihre FastLoad Teradata-Jobskripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten in verschiedene Datenbanken zu laden.

Bevor Sie ein FastLoad Job-Skript mit Substitutionsvariablen ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript UC4 (Automic) usw. verwenden.

AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihnen Werte zugewiesen haben. Bevor Sie mit der Konvertierung Ihrer FastLoad Teradata-Quelljobskripte beginnen, stellen Sie sicher, dass Sie allen Substitutionsvariablen Werte zuweisen. Sie können sie verwenden AWS SCT , um Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Um Substitutionsvariablen in Ihrem Jobskript zu konfigurieren FastLoad

1. Wenn Sie Ihrem AWS SCT Projekt Ihre FastLoad Teradata-Quelljobskripte hinzufügen, wählen Sie Substitutionsvariablen werden verwendet aus. Weitere Informationen zum Hinzufügen dieser Skripten finden Sie unter. Hinzufügen von FastLoad Jobskripten zu Ihrem Projekt AWS SCT
2. Geben Sie unter Variablenformat definieren einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

   Wenn die Namen Ihrer Substitutionsvariablen beispielsweise mit beginnen `${` und mit enden`}`, verwenden Sie den `\$\{\w+\}` regulären Ausdruck. Verwenden Sie den `\$\w+|\%\w+` regulären Ausdruck, um nach Substitutionsvariablen zu suchen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen.

   Reguläre Ausdrücke, die der Java-Syntax für reguläre Ausdrücke AWS SCT entsprechen. Weitere Informationen finden Sie unter java.util.regex Class Pattern in der Java-Dokumentation.
3. Wählen Sie OK, um Skripten in Ihr AWS SCT Projekt zu laden, und klicken Sie dann auf OK, um das Fenster Skripten laden zu schließen.
4. Erweitern Sie im linken Bereich den Knoten Scripts. Wählen Sie FastLoadund wählen Sie dann Ihren Ordner mit Skripten aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann unter Substitutionsvariablen die Option Variablen exportieren.

Sie können auch Substitutionsvariablen für ein Skript exportieren. Erweitern Sie Ihren Ordner mit Skripten, wählen Sie Ihr Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie unter Substitutionsvariablen die Option Variablen exportieren.

5. Geben Sie den Namen der Datei mit kommagetrennten Werten (CSV) ein, in der die Substitutionsvariablen gespeichert werden sollen, und wählen Sie dann Speichern.
6. Öffnen Sie diese CSV-Datei und geben Sie die Werte für die Substitutionsvariablen ein.

   AWS SCT Verwendet je nach Betriebssystem unterschiedliche Formate für die CSV-Datei. Die Werte in der Datei können entweder in Anführungszeichen eingeschlossen sein oder nicht. Stellen Sie sicher, dass Sie für die Werte der Substitutionsvariablen dasselbe Format wie für die anderen Werte in der Datei verwenden. AWS SCT kann die CSV-Datei mit Werten in verschiedenen Formaten nicht importieren.
7. Speichern Sie die CSV-Datei.
8. Erweitern Sie im linken Bereich den Knoten Scripts. Wählen Sie FastLoadund wählen Sie dann Ihr Skript aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann unter Substitutionsvariablen die Option Variablen importieren.
9. Wählen Sie Ihre CSV-Datei und dann Öffnen aus.
10. Wählen Sie Variablen, um alle erkannten Substitutionsvariablen und ihre Werte anzuzeigen.

FastLoad Teradata-Jobskripte konvertieren mit AWS SCT

Im Folgenden erfahren Sie, wie Sie FastLoad Teradata-Jobs mithilfe von Amazon Redshift RSQL konvertieren. AWS SCT

So konvertieren Sie ein FastLoad Teradata-Jobskript in Amazon Redshift RSQL

1. Fügen Sie Ihre FastLoad Jobskripte zu Ihrem Projekt hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von FastLoad Jobskripten zu Ihrem AWS SCT Projekt.
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten FastLoad .
3. Erweitern Sie im linken Bereich den Knoten Scripts.
4. Führen Sie eine der folgenden Aktionen aus:

- Um ein einzelnes FastLoad Job-Skript zu konvertieren, erweitern Sie den FastLoadKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Skript konvertieren.
- Um mehrere Skripten zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripten auswählen. Wählen Sie FastLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skript konvertieren. Führen Sie dann einen der folgenden Schritte aus:
  - Wenn Sie Ihre Quelldatendatei auf Amazon S3 speichern, wählen Sie den S3-Objektpfad als Speicherort der Quelldatendatei.

    Geben Sie Werte für den Amazon S3 S3-Bucket-Ordner und den Amazon S3 S3-Bucket für die Manifest-Datei für Ihre Quelldatendatei ein.
  - Wenn Sie Ihre Quelldatendatei nicht auf Amazon S3 speichern, wählen Sie Hostadresse als Speicherort der Quelldatendatei.

    Geben Sie Werte für die URL oder IP-Adresse des Hosts, den Anmeldenamen des Host-Benutzers und den Amazon S3 S3-Bucket für die Manifestdatei für Ihre Quelldatendatei ein.

5. Wählen Sie OK aus.

   AWS SCT konvertiert alle Ihre ausgewählten FastLoad Teradata-Jobskripte in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Suchen Sie Ihre konvertierten Skripten im Knoten „Skripte“ im Bereich „Zieldatenbank“.
6. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihrer konvertierten FastLoad Jobskripte.

Verwaltung von FastLoad Teradata-Jobskripten mit AWS SCT

Sie können mehrere FastLoad Teradata-Jobskripte hinzufügen oder ein FastLoad Jobskript aus Ihrem Projekt entfernen. AWS SCT

Um Ihrem Projekt ein neues FastLoad Jobskript hinzuzufügen AWS SCT

1. Erweitern Sie den Knoten Skripte im linken Bereich.
2. Wählen Sie den FastLoadKnoten aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripten laden.

4. Geben Sie die Informationen ein, die zum Hinzufügen eines neuen FastLoad Job-Skripts und zum Konfigurieren von Substitutionsvariablen erforderlich sind. Weitere Informationen erhalten Sie unter Hinzufügen von FastLoad Jobskripten zu Ihrem AWS SCT Projekt und Konfiguration von Substitutionsvariablen in Jobskripten FastLoad .

Um ein FastLoad Jobskript aus Ihrem Projekt zu AWS SCT entfernen

1. Erweitern Sie den FastLoadKnoten unter Skripte im linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen Sie einen Bewertungsbericht für die Konvertierung eines FastLoad Teradata-Jobskripts mit AWS SCT

Der Bewertungsbericht zur FastLoad Jobskriptkonvertierung enthält Informationen zur Konvertierung der FastLoad Befehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten in ein mit Amazon Redshift RSQL kompatibles Format. Der Bewertungsbericht enthält Aktionspunkte für FastLoad Befehle und SQL-Anweisungen, die nicht konvertiert AWS SCT werden können.

Um einen Bewertungsbericht zur Skriptkonvertierung für einen FastLoad Teradata-Job zu erstellen

1. Erweitern Sie den FastLoadKnoten unter Skripten im linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Bericht erstellen.
3. Rufen Sie die Registerkarte „Zusammenfassung“ auf.

   Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Bewertungsbericht für das FastLoad Jobskript angezeigt. Sie enthält Konvertierungsergebnisse für alle FastLoad Befehle und SQL-Anweisungen aus Ihren Quellskripten.
4. (Optional) Speichern Sie eine lokale Kopie des Bewertungsberichts zur FastLoad Jobskriptkonvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Um den Bewertungsbericht zur Konvertierung von FastLoad Jobskripten als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Skriptkonvertierung.

- Um den Bewertungsbericht zur FastLoad Jobskriptkonvertierung als CSV-Datei zu speichern, wählen Sie oben rechts die Option In CSV speichern.

  Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.

5. Wählen Sie die Registerkarte Aktionspunkte aus. Diese Registerkarte enthält eine Liste von Elementen, die manuell in Amazon Redshift RSQL konvertiert werden müssen. Wenn Sie ein Aktionselement aus der Liste auswählen, wird das Element aus Ihrem FastLoad Quell-Jobskript AWS SCT hervorgehoben, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten FastLoad Teradata-Jobskripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT Projekts bearbeiten. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie den Knoten RSQL-Skripten unter Skripten im Bereich Zieldatenbank.
2. Wählen Sie das konvertierte Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie Speichern.

   AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Konvertieren von MultiLoad Skripten nach Amazon Redshift RSQL mit AWS Schema Conversion Tool

Sie können AWS SCT es verwenden, um MultiLoad Teradata-Jobskripte in Amazon Redshift RSQL zu konvertieren.

Ein MultiLoad Teradata-Jobskript besteht aus einer Reihe von Befehlen für die Batch-Wartung Ihrer Teradata-Datenbank. Eine MultiLoad Teradata-Importaufgabe führt eine Reihe verschiedener Einfüge-, Aktualisierungs- und Löschvorgänge für bis zu fünf verschiedene Tabellen und Ansichten aus. MultiLoad Teradata-Löschaufgaben können eine große Anzahl von Zeilen aus einer einzelnen Tabelle entfernen.

AWS SCT konvertiert MultiLoad Teradata-Befehle und SQL-Anweisungen in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, verwenden Sie diese konvertierten Skripts, um Daten in Ihrer Amazon Redshift Redshift-Datenbank zu verwalten.

Themen

- Hinzufügen von MultiLoad Jobskripten zu Ihrem Projekt AWS SCT
- Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit MultiLoad AWS SCT
- MultiLoad Teradata-Jobskripte konvertieren mit AWS SCT
- Verwaltung von MultiLoad Teradata-Jobskripten mit AWS SCT
- Erstellen Sie einen Bewertungsbericht für die Konvertierung eines MultiLoad Teradata-Jobskripts mit AWS SCT
- Bearbeiten und Speichern Ihrer konvertierten MultiLoad Teradata-Jobskripte mit AWS SCT

Hinzufügen von MultiLoad Jobskripten zu Ihrem Projekt AWS SCT

Sie können einem einzelnen AWS SCT Projekt mehrere Skripte hinzufügen.

Um Ihrem AWS SCT Projekt ein MultiLoad Jobskript hinzuzufügen

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter the section called “Projekte starten und verwalten”.
2. Wählen Sie im Menü die Option Quelle hinzufügen und wählen Sie dann Teradata, um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter Teradata-Datenbanken.
3. Wählen Sie im Menü die Option Ziel hinzufügen aus, um Ihrem AWS SCT Projekt eine Amazon Redshift Redshift-Zieldatenbank hinzuzufügen.

   Sie können eine virtuelle Amazon Redshift Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift Redshift-Ziel umfasst. Weitere Informationen finden Sie unter Zuordnung neuer Datentypen in AWS Schema Conversion Tool.
5. Wählen Sie im Menü Ansicht die Option Hauptansicht.
6. Erweitern Sie im linken Bereich den Knoten Scripts.
7. Wählen Sie MultiLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skripten laden.
8. Geben Sie den Speicherort Ihrer MultiLoad Teradata-Quelljobskripte ein und wählen Sie Ordner auswählen aus.

   AWS SCT zeigt das Fenster Skripten laden an.
9. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Ihre MultiLoad Teradata-Jobskripte die Substitutionsvariablen nicht enthalten, wählen Sie Keine Substitutionsvariablen und dann OK, um Ihrem Projekt Skripten hinzuzufügen. AWS SCT
   - Wenn Ihre MultiLoad Teradata-Jobskripte die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten MultiLoad .

Konfiguration von Substitutionsvariablen in Teradata-Jobskripten mit MultiLoad AWS SCT

Ihre MultiLoad Teradata-Jobskripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten in verschiedene Datenbanken zu laden.

Bevor Sie ein MultiLoad Job-Skript mit Substitutionsvariablen ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript UC4 (Automic) usw. verwenden.

AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihnen Werte zugewiesen haben. Bevor Sie mit der Konvertierung Ihrer MultiLoad Teradata-Quelljobskripte beginnen, stellen Sie sicher, dass Sie allen Substitutionsvariablen Werte zugewiesen haben.
Sie können sie verwenden AWS SCT , um Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Um Substitutionsvariablen in Ihrem Jobskript zu konfigurieren MultiLoad

1. Wenn Sie Ihrem AWS SCT Projekt Ihre MultiLoad Teradata-Quelljobskripte hinzufügen, wählen Sie Substitutionsvariablen werden verwendet aus. Weitere Informationen zum Hinzufügen dieser Skripten finden Sie unter. Hinzufügen von MultiLoad Jobskripten zu Ihrem Projekt AWS SCT
2. Geben Sie unter Variablenformat definieren einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

   Wenn die Namen Ihrer Substitutionsvariablen beispielsweise mit beginnen `${` und mit enden`}`, verwenden Sie den `\$\{\w+\}` regulären Ausdruck. Verwenden Sie den `\$\w+|\%\w+` regulären Ausdruck, um nach Substitutionsvariablen zu suchen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen.

   Reguläre Ausdrücke, die der Java-Syntax für reguläre Ausdrücke AWS SCT entsprechen. Weitere Informationen finden Sie unter java.util.regex Class Pattern in der Java-Dokumentation.
3. Wählen Sie OK, um Skripten in Ihr AWS SCT Projekt zu laden, und klicken Sie dann auf OK, um das Fenster Skripten laden zu schließen.
4. Wählen Sie Variablen, um alle erkannten Substitutionsvariablen und ihre Werte anzuzeigen.
5. Geben Sie unter Wert den Wert für die Substitutionsvariable ein.

MultiLoad Teradata-Jobskripte konvertieren mit AWS SCT

Im Folgenden erfahren Sie, wie Sie MultiLoad Teradata-Jobs mithilfe von Amazon Redshift RSQL konvertieren. AWS SCT

So konvertieren Sie ein MultiLoad Teradata-Jobskript in Amazon Redshift RSQL

1. Fügen Sie Ihre MultiLoad Jobskripte zu Ihrem Projekt hinzu. AWS SCT Weitere Informationen finden Sie unter Hinzufügen von MultiLoad Jobskripten zu Ihrem AWS SCT Projekt.
2. Konfigurieren Sie die Substitutionsvariablen und geben Sie ihre Werte ein. Weitere Informationen finden Sie unter Konfiguration von Substitutionsvariablen in Jobskripten MultiLoad .
3. Erweitern Sie im linken Bereich den Knoten Scripts.
4. Führen Sie eine der folgenden Aktionen aus:

   - Um ein einzelnes MultiLoad Job-Skript zu konvertieren, erweitern Sie den MultiLoadKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Skript konvertieren.

- Um mehrere Skripten zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripten auswählen. Wählen Sie MultiLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Skript konvertieren.

5. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie Ihre Quelldatendatei auf Amazon S3 speichern, wählen Sie den S3-Objektpfad als Speicherort der Quelldatendatei.

     Geben Sie den Amazon S3 S3-Bucket-Ordner und den Amazon S3 S3-Bucket als Manifest-Datei für Ihre Quelldatendatei ein.

   - Wenn Sie Ihre Quelldatendatei nicht auf Amazon S3 speichern, wählen Sie Hostadresse als Speicherort der Quelldatendatei.

     Geben Sie die URL oder IP-Adresse des Hosts, den Anmeldenamen des Host-Benutzers und den Amazon S3 S3-Bucket für die Manifestdatei für Ihre Quelldatendatei ein.

6. Wählen Sie OK aus.

   AWS SCT konvertiert alle Ihre ausgewählten MultiLoad Teradata-Jobskripte in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Suchen Sie Ihre konvertierten Skripten im Knoten „Skripte“ im Bereich „Zieldatenbank“.

7. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihrer konvertierten MultiLoad Jobskripte.

Verwaltung von MultiLoad Teradata-Jobskripten mit AWS SCT

Sie können mehrere MultiLoad Teradata-Jobskripte hinzufügen oder ein MultiLoad Jobskript aus Ihrem Projekt entfernen. AWS SCT

Um Ihrem Projekt ein neues MultiLoad Jobskript hinzuzufügen AWS SCT

1. Erweitern Sie den Knoten Skripte im linken Bereich.
2. Wählen Sie den MultiLoadKnoten aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripten laden.
4. Geben Sie die Informationen ein, die zum Hinzufügen eines neuen MultiLoad Job-Skripts und zum Konfigurieren von Substitutionsvariablen erforderlich sind. Weitere Informationen erhalten

Sie unter Hinzufügen von MultiLoad Jobskripten zu Ihrem AWS SCT Projekt und Konfiguration von Substitutionsvariablen in Jobskripten MultiLoad .

Um ein MultiLoad Jobskript aus Ihrem Projekt zu AWS SCT entfernen

1. Erweitern Sie den MultiLoadKnoten unter Skripte im linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen Sie einen Bewertungsbericht für die Konvertierung eines MultiLoad Teradata-Jobskripts mit AWS SCT

Der Bewertungsbericht zur MultiLoad Jobskriptkonvertierung enthält Informationen zur Konvertierung der MultiLoad Befehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten in Amazon Redshift RSQL-Befehle und SQL-Anweisungen für Amazon Redshift. Der Bewertungsbericht enthält Aktionspunkte für MultiLoad Befehle und SQL-Anweisungen, die nicht konvertiert werden AWS SCT können.

Um einen Bewertungsbericht zur Skriptkonvertierung für einen MultiLoad Teradata-Job zu erstellen

1. Erweitern Sie den MultiLoadKnoten unter Skripten im linken Bereich.
2. Wählen Sie die Skripts aus, für die der Bewertungsbericht erstellt werden soll, öffnen Sie das Kontextmenü (mit der rechten Maustaste) und wählen Sie dann Bericht erstellen.
3. Rufen Sie die Registerkarte „Zusammenfassung“ auf. Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Bewertungsbericht für das MultiLoad Jobskript angezeigt. Sie enthält Konvertierungsergebnisse für alle MultiLoad Befehle und SQL-Anweisungen aus Ihren Quellskripten.
4. (Optional) Speichern Sie eine lokale Kopie des Bewertungsberichts zur MultiLoad Jobskriptkonvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Um den Bewertungsbericht zur Konvertierung von MultiLoad Jobskripten als PDF-Datei zu speichern, wählen Sie oben rechts die Option Als PDF speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um den Bewertungsbericht zur Konvertierung von MultiLoad Jobskripten als CSV-Dateien zu speichern, wählen Sie oben rechts die Option In CSV speichern.

  AWS SCT erstellt zwei CSV-Dateien. Diese Dateien enthalten die Zusammenfassung, Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung der Skripts erforderlich ist.

5. Wählen Sie die Registerkarte Aktionspunkte. Diese Registerkarte enthält eine Liste von Elementen, die manuell in Amazon Redshift RSQL konvertiert werden müssen. Wenn Sie ein Aktionselement aus der Liste auswählen, wird das Element aus Ihrem MultiLoad Quell-Jobskript AWS SCT hervorgehoben, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten MultiLoad Teradata-Jobskripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT Projekts bearbeiten. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie den Knoten RSQL-Skripten unter Skripten im Bereich Zieldatenbank.
2. Wählen Sie das konvertierte Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie Speichern.

   AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Migration von Big-Data-Frameworks mit AWS Schema Conversion Tool

Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um Big-Data-Frameworks auf das zu migrieren. AWS Cloud

AWS SCT Unterstützt derzeit die Migration von Hadoop-Clustern zu Amazon EMR und Amazon S3. Dieser Migrationsprozess umfasst Hive- und HDFS-Dienste.

Außerdem können Sie AWS SCT damit die Konvertierung Ihrer Apache Oozie-Orchestrierungs-Workflows zu automatisieren. AWS Step Functions

Themen

- Migration von Hadoop-Workloads zu Amazon EMR mit AWS Schema Conversion Tool
- Konvertierung von Oozie-Workflows in mit AWS Step FunctionsAWS Schema Conversion Tool

Migration von Hadoop-Workloads zu Amazon EMR mit AWS Schema Conversion Tool

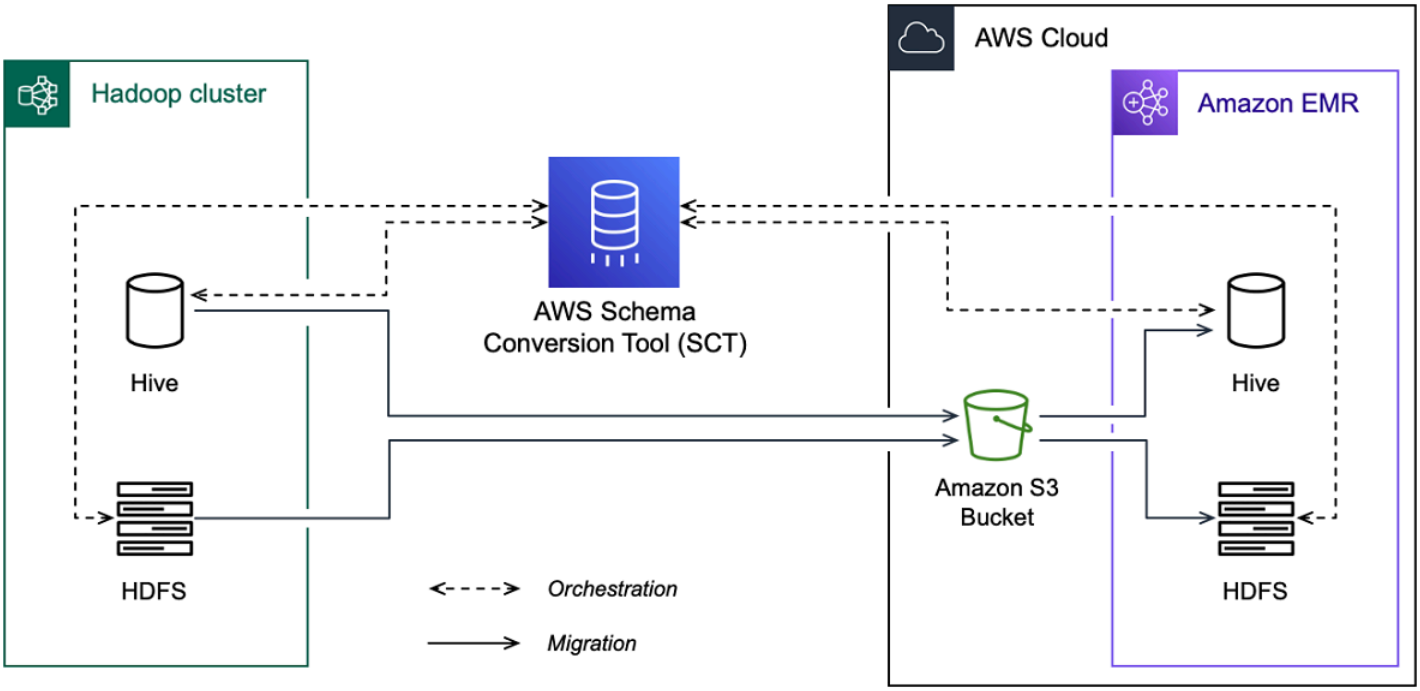
Um Apache Hadoop-Cluster zu migrieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.670 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle (CLI) von vertraut AWS SCT. Weitere Informationen finden Sie unter CLI-Referenz für AWS Schema Conversion Tool.

Themen

- Überblick über die Migration
- Schritt 1: Connect zu Ihren Hadoop-Clustern her
- Schritt 2: Richten Sie die Zuordnungsregeln ein
- Schritt 3: Erstellen Sie einen Bewertungsbericht
- Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR mit AWS SCT
- Ausführen Ihres CLI-Skripts
- Verwaltung Ihres Big-Data-Migrationsprojekts

Überblick über die Migration

Die folgende Abbildung zeigt das Architekturdiagramm der Migration von Apache Hadoop zu Amazon EMR.



AWS SCT migriert Daten und Metadaten von Ihrem Hadoop-Quell-Cluster in einen Amazon S3 S3-Bucket. Als Nächstes werden Ihre Hive-Quellmetadaten AWS SCT verwendet, um Datenbankobjekte im Amazon EMR Hive-Zielservice zu erstellen. Optional können Sie Hive so konfigurieren, dass es als Metastore verwendet. AWS Glue Data Catalog In diesem Fall AWS SCT migriert Ihre Hive-Quellmetadaten in den. AWS Glue Data Catalog

Anschließend können AWS SCT Sie die Daten von einem Amazon S3-Bucket zu Ihrem Amazon EMR HDFS-Zielservice migrieren. Alternativ können Sie die Daten in Ihrem Amazon S3 S3-Bucket belassen und als Datenspeicher für Ihre Hadoop-Workloads verwenden.

Um die Hapood-Migration zu starten, erstellen Sie Ihr AWS SCT CLI-Skript und führen es aus. Dieses Skript enthält den kompletten Befehlssatz zur Ausführung der Migration. Sie können eine Vorlage des Hadoop-Migrationsskripts herunterladen und bearbeiten. Weitere Informationen finden Sie unter CLI-Szenarien abrufen.

Stellen Sie sicher, dass Ihr Skript die folgenden Schritte umfasst, damit Sie Ihre Migration von Apache Hadoop zu Amazon S3 und Amazon EMR ausführen können.

Schritt 1: Connect zu Ihren Hadoop-Clustern her

Um die Migration Ihres Apache Hadoop-Clusters zu starten, erstellen Sie ein neues Projekt. AWS SCT Stellen Sie als Nächstes eine Verbindung zu Ihren Quell- und Zielclustern her. Stellen Sie sicher, dass Sie Ihre AWS Zielressourcen erstellen und bereitstellen, bevor Sie mit der Migration beginnen.

In diesem Schritt verwenden Sie die folgenden AWS SCT CLI-Befehle.

- `CreateProject`— um ein neues AWS SCT Projekt zu erstellen.
- `AddSourceCluster`— um eine Verbindung zum Quell-Hadoop-Cluster in Ihrem AWS SCT Projekt herzustellen.
- `AddSourceClusterHive`— um eine Verbindung zum Quell-Hive-Dienst in Ihrem Projekt herzustellen.
- `AddSourceClusterHDFS`— um eine Verbindung zum Quell-HDFS-Dienst in Ihrem Projekt herzustellen.
- `AddTargetCluster`— um eine Verbindung zum Amazon EMR-Zielcluster in Ihrem Projekt herzustellen.
- `AddTargetClusterS3`— um den Amazon S3 S3-Bucket zu Ihrem Projekt hinzuzufügen.
- `AddTargetClusterHive`— um eine Verbindung zum Ziel-Hive-Dienst in Ihrem Projekt herzustellen
- `AddTargetClusterHDFS`— um eine Verbindung zum Ziel-HDFS-Dienst in Ihrem Projekt herzustellen

Beispiele für die Verwendung dieser AWS SCT CLI-Befehle finden Sie unterVerbindung zu Apache Hadoop herstellen.

Wenn Sie den Befehl ausführen, der eine Verbindung zu einem Quell- oder Zielcluster herstellt, AWS SCT versucht er, die Verbindung zu diesem Cluster herzustellen. Wenn der Verbindungsversuch fehlschlägt, AWS SCT beendet es die Ausführung der Befehle aus Ihrem CLI-Skript und zeigt eine Fehlermeldung an.

Schritt 2: Richten Sie die Zuordnungsregeln ein

Nachdem Sie eine Verbindung zu Ihren Quell- und Zielclustern hergestellt haben, richten Sie die Zuordnungsregeln ein. Eine Zuordnungsregel definiert das Migrationsziel für einen Quellcluster.

Stellen Sie sicher, dass Sie Zuordnungsregeln für alle Quellcluster einrichten, die Sie Ihrem AWS SCT Projekt hinzugefügt haben. Weitere Informationen zu Zuordnungsregeln finden Sie unterZuordnung von Datentypen in der AWS Schema Conversion Tool.

In diesem Schritt verwenden Sie den `AddServerMapping` Befehl. Dieser Befehl verwendet zwei Parameter, die die Quell- und Zielcluster definieren. Sie können den `AddServerMapping` Befehl mit dem expliziten Pfad zu Ihren Datenbankobjekten oder mit einem Objektnamen verwenden. Bei der ersten Option geben Sie den Typ des Objekts und seinen Namen an. Bei der zweiten Option geben Sie nur die Objektnamen an.

- `sourceTreePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten.

  `targetTreePath`— der explizite Pfad zu Ihren Zieldatenbankobjekten.
- `sourceNamePath`— der Pfad, der nur die Namen Ihrer Quellobjekte enthält.

  `targetNamePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält.

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die explizite Pfade für die `testdb` Hive-Quelldatenbank und den EMR-Zielcluster verwendet.

```
AddServerMapping
 -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
 -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden. Um die CLI-Befehle unter Linux auszuführen, stellen Sie sicher, dass Sie die Dateipfade entsprechend für Ihr Betriebssystem aktualisiert haben.

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die die Pfade verwendet, die nur die Objektnamen enthalten.

```
AddServerMapping
 -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
 -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Sie können Amazon EMR oder Amazon S3 als Ziel für Ihr Quellobjekt wählen. Für jedes Quellobjekt können Sie nur ein Ziel in einem einzigen AWS SCT Projekt auswählen. Um das Migrationsziel

für ein Quellobjekt zu ändern, löschen Sie die bestehende Zuordnungsregel und erstellen Sie dann eine neue Zuordnungsregel. Verwenden Sie den `DeleteServerMapping` Befehl, um eine Zuordnungsregel zu löschen. Dieser Befehl verwendet einen der beiden folgenden Parameter.

- `sourceTreePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten.
- `sourceNamePath`— der Pfad, der nur die Namen Ihrer Quellobjekte enthält.

Weitere Informationen zu den `DeleteServerMapping` Befehlen `AddServerMapping` und finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Schritt 3: Erstellen Sie einen Bewertungsbericht

Bevor Sie mit der Migration beginnen, empfehlen wir, einen Bewertungsbericht zu erstellen. In diesem Bericht werden alle Migrationsaufgaben zusammengefasst und die während der Migration zu erwartenden Maßnahmen detailliert beschrieben. Um sicherzustellen, dass Ihre Migration nicht fehlschlägt, lesen Sie diesen Bericht und gehen Sie vor der Migration auf die Aktionspunkte ein. Weitere Informationen finden Sie unter Bewertungsbericht.

In diesem Schritt verwenden Sie den `CreateMigrationReport` Befehl. Dieser Befehl verwendet zwei Parameter. Der `treePath` Parameter ist obligatorisch und der `forceMigrate` Parameter ist optional.

- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceMigrate`— Wenn diese Option auf gesetzt ist`true`, AWS SCT wird die Migration auch dann fortgesetzt, wenn Ihr Projekt einen HDFS-Ordner und eine Hive-Tabelle enthält, die auf dasselbe Objekt verweisen. Der Standardwert ist `false`.

Anschließend können Sie eine Kopie des Bewertungsberichts als PDF- oder CSV-Datei (Comma-Separated Value) speichern. Verwenden Sie dazu den Befehl `SaveReportPDF` oder`SaveReportCSV`.

Der `SaveReportPDF` Befehl speichert eine Kopie Ihres Bewertungsberichts als PDF-Datei. Dieser Befehl verwendet vier Parameter. Der `file` Parameter ist obligatorisch, andere Parameter sind optional.

- `file`— der Pfad zur PDF-Datei und ihr Name.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen der Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Der `SaveReportCSV` Befehl speichert Ihren Bewertungsbericht in drei CSV-Dateien. Dieser Befehl verwendet vier Parameter. Der `directory` Parameter ist obligatorisch, andere Parameter sind optional.

- `directory`— der Pfad zu dem Ordner, in dem die CSV-Dateien AWS SCT gespeichert werden.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen der Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts in der `c:\sct\ar.pdf` Datei gespeichert.

```
SaveReportPDF
 -file:'c:\sct\ar.pdf'
/
```

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts als CSV-Dateien im `c:\sct` Ordner gespeichert.

```
SaveReportCSV
 -file:'c:\sct'
/
```

Weitere Informationen zu den `SaveReportCSV` Befehlen `SaveReportPDF` und finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR mit AWS SCT

Nachdem Sie Ihr AWS SCT Projekt konfiguriert haben, starten Sie die Migration Ihres lokalen Apache Hadoop-Clusters zum. AWS Cloud

In diesem Schritt verwenden Sie die Befehle `MigrateMigrationStatus`, und`ResumeMigration`.

Der `Migrate` Befehl migriert Ihre Quellobjekte in den Zielcluster. Dieser Befehl verwendet vier Parameter. Stellen Sie sicher, dass Sie den `treePath` Parameter `filter` oder angeben. Andere Parameter sind optional.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Bereich Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceLoad`— Wenn diese Option auf gesetzt ist`true`, AWS SCT werden Datenbank-Metadatenbäume während der Migration automatisch geladen. Der Standardwert ist `false`.
- `forceMigrate`— Wenn diese Option auf gesetzt ist`true`, AWS SCT wird die Migration auch dann fortgesetzt, wenn Ihr Projekt einen HDFS-Ordner und eine Hive-Tabelle enthält, die auf dasselbe Objekt verweisen. Der Standardwert ist `false`.

Der `MigrationStatus` Befehl gibt Informationen über den Migrationsfortschritt zurück. Um diesen Befehl auszuführen, geben Sie den Namen Ihres Migrationsprojekts für den `name` Parameter ein. Sie haben diesen Namen im `CreateProject` Befehl angegeben.

Der `ResumeMigration` Befehl setzt die unterbrochene Migration fort, die Sie mit dem `Migrate` Befehl gestartet haben. Der `ResumeMigration` Befehl verwendet keine Parameter. Um die Migration fortzusetzen, müssen Sie eine Verbindung zu Ihren Quell- und Zielclustern herstellen. Weitere Informationen finden Sie unter Verwaltung Ihres Migrationsprojekts.

Das folgende Codebeispiel migriert Daten von Ihrem HDFS-Quellservice zu Amazon EMR.

```
Migrate
 -treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
 -forceMigrate: 'true'
/
```

Ausführen Ihres CLI-Skripts

Wenn Sie mit der Bearbeitung Ihres AWS SCT CLI-Skripts fertig sind, speichern Sie es als Datei mit der `.scts` Erweiterung. Jetzt können Sie Ihr Skript im `app` Ordner Ihres AWS SCT Installationspfads ausführen. Führen Sie dazu den folgenden Befehl aus.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Ersetzen Sie im vorherigen Beispiel durch *`script_path`* den Pfad zu Ihrer Datei mit dem CLI-Skript. Weitere Hinweise zum Ausführen von CLI-Skripts in AWS SCT finden Sie unterSkriptmodus.

Verwaltung Ihres Big-Data-Migrationsprojekts

Nachdem Sie die Migration abgeschlossen haben, können Sie Ihr AWS SCT Projekt für die future Verwendung speichern und bearbeiten.

Verwenden Sie den `SaveProject` Befehl, um Ihr AWS SCT Projekt zu speichern. Dieser Befehl verwendet keine Parameter.

Das folgende Codebeispiel speichert Ihr AWS SCT Projekt.

```
SaveProject
/
```

Verwenden Sie den `OpenProject` Befehl, um Ihr AWS SCT Projekt zu öffnen. Dieser Befehl verwendet einen obligatorischen Parameter. Geben Sie für den `file` Parameter den Pfad zu Ihrer AWS SCT Projektdatei und deren Namen ein. Sie haben den Projektnamen im `CreateProject` Befehl angegeben. Stellen Sie sicher, dass Sie dem Namen Ihrer Projektdatei die `.scts` Erweiterung hinzufügen, um den `OpenProject` Befehl auszuführen.

Im folgenden Codebeispiel wird das `hadoop_emr` Projekt aus dem `c:\sct` Ordner geöffnet.

```
OpenProject
 -file: 'c:\sct\hadoop_emr.scts'
/
```

Nachdem Sie Ihr AWS SCT Projekt geöffnet haben, müssen Sie die Quell- und Zielcluster nicht mehr hinzufügen, da Sie sie bereits zu Ihrem Projekt hinzugefügt haben. Um mit Ihren Quell- und Zielclustern arbeiten zu können, müssen Sie eine Verbindung zu ihnen herstellen. Dazu verwenden Sie die `ConnectTargetCluster` Befehle `ConnectSourceCluster` und. Diese

Befehle verwenden dieselben Parameter wie die `AddTargetCluster` Befehle `AddSourceCluster` und. Sie können Ihr CLI-Skript bearbeiten und den Namen dieser Befehle ersetzen, sodass die Liste der Parameter unverändert bleibt.

Das folgende Codebeispiel stellt eine Verbindung zum Quell-Hadoop-Cluster her.

```
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'
/
```

Das folgende Codebeispiel stellt eine Verbindung mit dem Amazon EMR-Zielcluster her.

```
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'
/
```

Ersetzen Sie es im vorherigen Beispiel *`hadoop_address`* durch die IP-Adresse Ihres Hadoop-Clusters. Konfigurieren Sie bei Bedarf den Wert der Port-Variablen. Ersetzen Sie als Nächstes *`hadoop_user`* und *`hadoop_password`* durch den Namen Ihres Hadoop-Benutzers und das Passwort für diesen Benutzer. Geben Sie für *`path\name`* den Namen und den Pfad zur PEM-Datei für Ihren Hadoop-Quellcluster ein. Weitere Informationen zum Hinzufügen Ihrer Quell- und

Zielcluster finden Sie unter. Herstellen einer Verbindung zu Apache Hadoop-Datenbanken mit dem AWS Schema Conversion Tool

Nachdem Sie eine Verbindung zu Ihren Quell- und Ziel-Hadoop-Clustern hergestellt haben, müssen Sie eine Verbindung zu Ihren Hive- und HDFS-Services sowie zu Ihrem Amazon S3 S3-Bucket herstellen. Dazu verwenden Sie die Befehle`ConnectSourceClusterHive`,, `ConnectSourceClusterHdfs` `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs`, und. `ConnectTargetClusterS3` Diese Befehle verwenden dieselben Parameter wie die Befehle, mit denen Sie Hive- und HDFS-Services sowie den Amazon S3 S3-Bucket zu Ihrem Projekt hinzugefügt haben. Bearbeiten Sie das CLI-Skript, um das `Add` Präfix durch `Connect` die Befehlsnamen zu ersetzen.

Konvertierung von Oozie-Workflows in mit AWS Step FunctionsAWS Schema Conversion Tool

Um Apache Oozie-Workflows zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.671 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle (CLI) von vertraut AWS SCT. Weitere Informationen finden Sie unter CLI-Referenz für AWS Schema Conversion Tool.

Themen

- Überblick über die Konvertierung
- Schritt 1: Connect zu Ihren Quell- und Zieldiensten her
- Schritt 2: Richten Sie die Zuordnungsregeln ein
- Schritt 3: Parameter konfigurieren
- Schritt 4: Erstellen eines Bewertungsberichts
- Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows in mit AWS Step FunctionsAWS SCT
- Ausführen Ihres CLI-Skripts
- Apache Oozie-Knoten, die konvertiert AWS SCT werden können in AWS Step Functions

Überblick über die Konvertierung

Ihr Apache Oozie-Quellcode umfasst Aktionsknoten, Kontrollflussknoten und Jobeigenschaften. Aktionsknoten definieren die Jobs, die Sie in Ihrem Apache Oozie-Workflow ausführen. Wenn Sie Apache Oozie verwenden, um Ihren Apache Hadoop-Cluster zu orchestrieren, enthält ein Aktionsknoten einen Hadoop-Job. Kontrollflussknoten bieten einen Mechanismus zur Steuerung

des Workflow-Pfads. Zu den Kontrollflussknoten gehören Knoten wie `start`,`end`,`decision`,`fork`, und`join`.

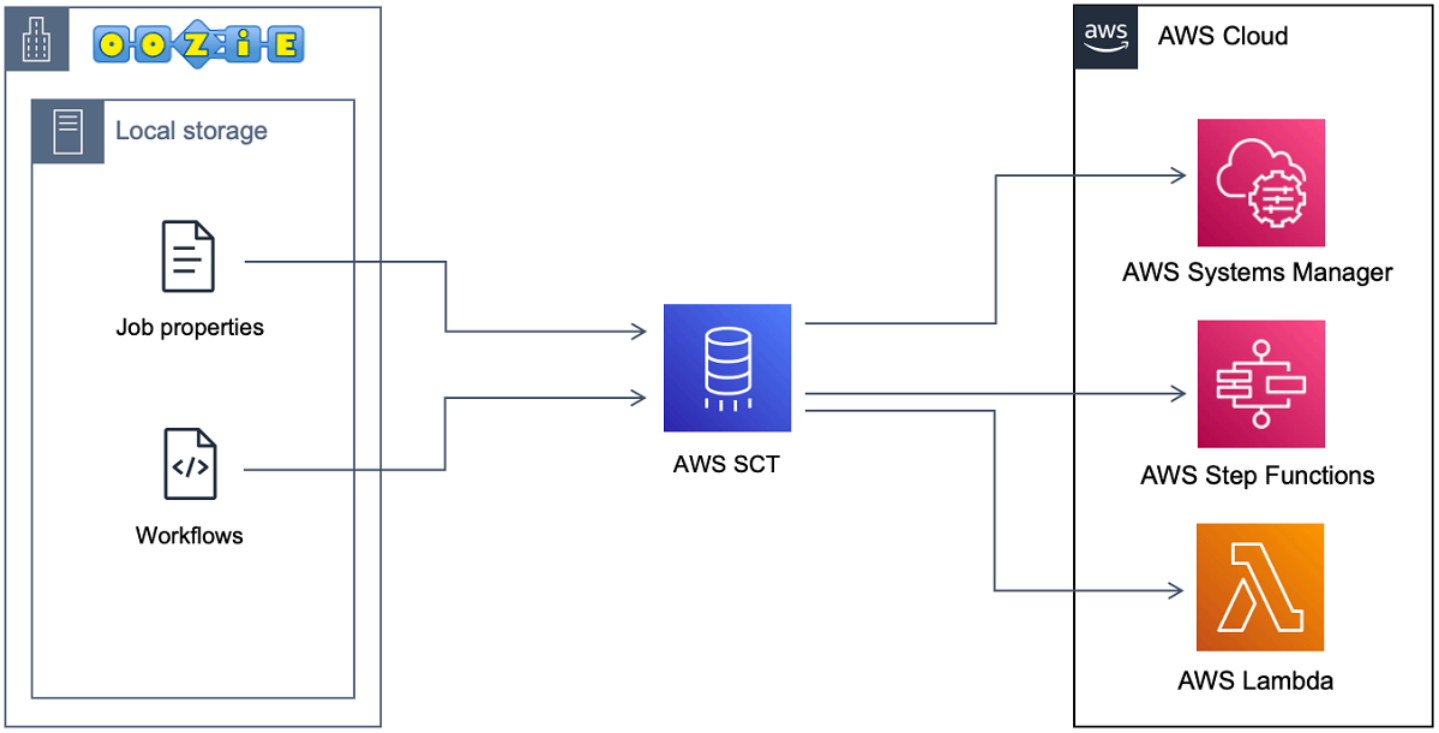
AWS SCT konvertiert Ihre Quell- und Kontrollflussknoten in AWS Step Functions. AWS Step Functions In definieren Sie Ihre Workflows in der Amazon States Language (ASL). AWS SCT verwendet ASL, um Ihre Zustandsmaschine zu definieren. Dabei handelt es sich um eine Sammlung von Zuständen, die Arbeit verrichten, festlegen, in welche Status als Nächstes übergegangen werden soll, mit einem Fehler beenden und so weiter. AWS SCT Lädt als Nächstes die JSON-Dateien mit Definitionen von Zustandsmaschinen hoch. Anschließend AWS SCT können Sie Ihre AWS Identity and Access Management (IAM-) Rolle verwenden, um Ihre Zustandsmaschinen in zu konfigurieren. AWS Step Functions Weitere Informationen finden Sie unter Was ist AWS Step Functions? im AWS Step Functions Entwicklerhandbuch.

AWS SCT Erstellt außerdem ein Erweiterungspaket mit AWS Lambda Funktionen, die die Quellfunktionen emulieren, die AWS Step Functions nicht unterstützt werden. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool.

AWS SCT migriert Ihre Quell-Jobeigenschaften zu. AWS Systems Manager Zum Speichern von Parameternamen und -werten AWS SCT verwendet Parameter Store, eine Funktion von AWS Systems Manager. Weitere Informationen finden Sie unter Was ist AWS Systems Manager? im AWS Systems Manager Benutzerhandbuch.

Sie können AWS SCT es verwenden, um die Werte und Namen Ihrer Parameter automatisch zu aktualisieren. Aufgrund der Architekturunterschiede zwischen Apache Oozie und AWS Step Functions müssen Sie möglicherweise Ihre Parameter konfigurieren. AWS SCT kann einen bestimmten Parameternamen oder -wert in Ihren Quelldateien finden und diese durch neue Werte ersetzen. Weitere Informationen finden Sie unter Schritt 3: Parameter konfigurieren.

Die folgende Abbildung zeigt das Architekturdiagramm der Apache Oozie-Konvertierung zu AWS Step Functions.



Um die Konvertierung zu starten, erstellen Sie Ihr AWS SCT CLI-Skript und führen Sie es aus. Dieses Skript enthält den vollständigen Befehlssatz zum Ausführen der Konvertierung. Sie können eine Vorlage des Apache Oozie-Konvertierungsskripts herunterladen und bearbeiten. Weitere Informationen finden Sie unter CLI-Szenarien abrufen.

Stellen Sie sicher, dass Ihr Skript die folgenden Schritte enthält.

Schritt 1: Connect zu Ihren Quell- und Zieldiensten her

Um die Konvertierung Ihres Apache Oozie-Clusters zu starten, erstellen Sie ein neues AWS SCT Projekt. Stellen Sie als Nächstes eine Verbindung zu Ihren Quell- und Zieldiensten her. Stellen Sie sicher, dass Sie Ihre AWS Zielressourcen erstellen und bereitstellen, bevor Sie mit der Migration beginnen. Weitere Informationen finden Sie unter Voraussetzungen für die Verwendung von Apache Oozie als Quelle.

In diesem Schritt verwenden Sie die folgenden AWS SCT CLI-Befehle.

- `CreateProject`— um ein neues AWS SCT Projekt zu erstellen.
- `AddSource`— um Ihre Apache Oozie-Quelldateien zu Ihrem AWS SCT Projekt hinzuzufügen.
- `ConnectSource`— um eine Verbindung zu Apache Oozie als Quelle herzustellen.
- `AddTarget`— um es Ihrem Projekt AWS Step Functions als Migrationsziel hinzuzufügen.

- `ConnectTarget`— um eine Verbindung herzustellen AWS Step Functions.

Beispiele für die Verwendung dieser AWS SCT CLI-Befehle finden Sie unterVerbindung zu Apache Oozie herstellen.

Wenn Sie die `ConnectTarget` Befehle `ConnectSource` oder ausführen, AWS SCT versucht es, die Verbindung zu Ihren Diensten herzustellen. Wenn der Verbindungsversuch fehlschlägt, AWS SCT beendet es die Ausführung der Befehle aus Ihrem CLI-Skript und zeigt eine Fehlermeldung an.

Schritt 2: Richten Sie die Zuordnungsregeln ein

Nachdem Sie eine Verbindung zu Ihren Quell- und Zieldiensten hergestellt haben, richten Sie die Zuordnungsregeln ein. Eine Zuordnungsregel definiert das Migrationsziel für Ihre Apache Oozie-Quell-Workflows und -Parameter. Weitere Informationen zu Zuordnungsregeln finden Sie unterZuordnung von Datentypen in der AWS Schema Conversion Tool.

Verwenden Sie den `AddServerMapping` Befehl, um Quell- und Zielobjekte für die Konvertierung zu definieren. Dieser Befehl verwendet zwei Parameter: `sourceTreePath` und`targetTreePath`. Die Werte dieser Parameter beinhalten einen expliziten Pfad zu Ihren Quell- und Zielobjekten. Damit Apache Oozie AWS Step Functions konvertiert werden kann, müssen diese Parameter mit `ETL` beginnen.

Das folgende Codebeispiel erstellt eine Zuordnungsregel für `AWS_STEP_FUNCTIONS` Objekte `OOZIE` und. Sie haben diese Objekte mit den `AddTarget` Befehlen `AddSource` und im vorherigen Schritt zu Ihrem AWS SCT Projekt hinzugefügt.

```
AddServerMapping
    -sourceTreePath: 'ETL.APACHE_OOZIE'
    -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Weitere Informationen zu dem `AddServerMapping` Befehl finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Schritt 3: Parameter konfigurieren

Wenn Ihre Apache Oozie-Quell-Workflows Parameter verwenden, müssen Sie deren Werte möglicherweise nach der Konvertierung in ändern. AWS Step Functions Außerdem müssen Sie möglicherweise neue Parameter hinzufügen, um sie mit Ihrem AWS Step Functions zu verwenden.

In diesem Schritt verwenden Sie die `AddTargetParameter` Befehle `AddParameterMapping` und.

Verwenden Sie den `AddParameterMapping` Befehl, um die Parameterwerte in Ihren Quelldateien zu ersetzen. AWS SCT scannt Ihre Quelldateien, findet die Parameter anhand ihres Namens oder Werts und ändert ihre Werte. Sie können einen einzigen Befehl ausführen, um alle Ihre Quelldateien zu scannen. Sie definieren den Umfang der zu scannenden Dateien anhand eines der ersten drei Parameter aus der folgenden Liste. Dieser Befehl verwendet bis zu sechs Parameter.

- `filterName`— der Name des Filters für Ihre Quellobjekte. Sie können einen Filter mit dem `CreateFilter` Befehl erstellen.
- `treePath`— der explizite Pfad zu Ihren Quellobjekten.
- `namePath`— der explizite Pfad zu einem bestimmten Quellobjekt.
- `sourceParameterName`— der Name Ihres Quellparameters.
- `sourceValue`— der Wert Ihres Quellparameters.
- `targetValue`— der Wert Ihres Zielparameters.

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Wert `c:\oozie\hive.py` dem `s3://bucket-oozie/hive.py` Wert entspricht.

```
AddParameterMapping
 -treePath: 'ETL.OOZIE.Applications'
 -sourceValue: 'c:\oozie\hive.py'
 -targetValue: 's3://bucket-oozie/hive.py'
/
```

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Name `nameNode` dem `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` Wert entspricht.

```
AddParameterMapping
    -treePath: 'ETL.OOZIE_SOURCE.Applications'
    -sourceParameter: 'nameNode'
    -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Name gleich `nameNode` und der Wert `hdfs://ip-55.eu-west-1.compute.internal:8020` dem Wert aus dem `targetValue` Parameter entspricht.

```
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'
/
```

Verwenden Sie den `AddTargetParameter` Befehl, um Ihren Zieldateien zusätzlich zu einem vorhandenen Parameter aus Ihren Quelldateien einen neuen Parameter hinzuzufügen. Dieser Befehl verwendet denselben Parametersatz wie der `AddParameterMapping` Befehl.

Im folgenden Codebeispiel `clusterId` wird der Zielparameter anstelle des `nameNode` Parameters hinzugefügt.

```
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'
/
```

Weitere Informationen zu den `CreateFilter` Befehlen `AddServerMappingAddParameterMapping`,`AddTargetParameter`, und finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Schritt 4: Erstellen eines Bewertungsberichts

Bevor Sie mit der Konvertierung beginnen, empfehlen wir, einen Bewertungsbericht zu erstellen. Dieser Bericht fasst alle Migrationsaufgaben zusammen und beschreibt detailliert die Aktionspunkte, die sich während der Migration ergeben werden. Um sicherzustellen, dass Ihre Migration nicht fehlschlägt, lesen Sie diesen Bericht und gehen Sie vor der Migration auf die Aktionspunkte ein. Weitere Informationen finden Sie unter Bewertungsbericht.

In diesem Schritt verwenden Sie den `CreateReport` Befehl. Dieser Befehl verwendet zwei Parameter. Der erste Parameter beschreibt die Quellobjekte, für die ein Bewertungsbericht AWS SCT erstellt wird. Verwenden Sie dazu einen der folgenden Parameter:`filterName`,`treePath`, oder`namePath`. Dieser Parameter ist obligatorisch. Sie können auch einen optionalen booleschen Parameter hinzufügen. `forceLoad` Wenn Sie diesen Parameter auf setzen`true`, AWS SCT werden

automatisch alle untergeordneten Objekte für das Quellobjekt geladen, das Sie im Befehl angeben. `CreateReport`

Im folgenden Codebeispiel wird ein Bewertungsbericht für den `Applications` Knoten Ihrer Oozie-Quelldateien erstellt.

```
CreateReport
    -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Anschließend können Sie eine Kopie des Bewertungsberichts als PDF- oder CSV-Datei (Comma-Separated Value) speichern. Verwenden Sie dazu den Befehl `SaveReportPDF` oder`SaveReportCSV`.

Der `SaveReportPDF` Befehl speichert eine Kopie Ihres Bewertungsberichts als PDF-Datei. Dieser Befehl verwendet vier Parameter. Der `file` Parameter ist obligatorisch, andere Parameter sind optional.

- `file`— der Pfad zur PDF-Datei und ihr Name.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen der Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Der `SaveReportCSV` Befehl speichert Ihren Bewertungsbericht in CSV-Dateien. Dieser Befehl verwendet vier Parameter. Der `directory` Parameter ist obligatorisch, andere Parameter sind optional.

- `directory`— der Pfad zu dem Ordner, in dem die CSV-Dateien AWS SCT gespeichert werden.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen der Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts in der `c:\sct\ar.pdf` Datei gespeichert.

```
SaveReportPDF
 -file:'c:\sct\ar.pdf'
/
```

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts als CSV-Dateien im `c:\sct` Ordner gespeichert.

```
SaveReportCSV
 -file:'c:\sct'
/
```

Weitere Informationen zu den `CreateReport` `SaveReportCSV` Befehlen `SaveReportPDF` und finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows in mit AWS Step FunctionsAWS SCT

Nachdem Sie Ihr AWS SCT Projekt konfiguriert haben, konvertieren Sie Ihren Quellcode und wenden Sie ihn auf den AWS Cloud an.

In diesem Schritt verwenden Sie die `ApplyToTarget` Befehle `Convert` `SaveOnS3ConfigureStateMachine`,, und.

Der `Migrate` Befehl migriert Ihre Quellobjekte in den Zielcluster. Dieser Befehl verwendet vier Parameter. Stellen Sie sicher, dass Sie den `treePath` Parameter `filter` oder angeben. Andere Parameter sind optional.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Bereich Ihrer zu migrierenden Quellobjekte zu definieren.
- `namePath`— der explizite Pfad zu einem bestimmten Quellobjekt.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceLoad`— Wenn diese Option auf gesetzt ist`true`, AWS SCT werden Datenbank-Metadatenbäume während der Migration automatisch geladen. Der Standardwert ist `false`.

Das folgende Codebeispiel konvertiert Dateien aus dem `Applications` Ordner in Ihren Oozie-Quelldateien.

```
Convert
    -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Der `SaveOnS3` lädt die Definitionen der Zustandsmaschinen in Ihren Amazon S3 S3-Bucket hoch. Dieser Befehl verwendet den `treePath` Parameter. Um diesen Befehl auszuführen, verwenden Sie den Zielordner mit den Definitionen von Zustandsmaschinen als Wert für diesen Parameter.

Im Folgenden wird der `State machine definitions` Ordner Ihres `AWS_STEP_FUNCTIONS` Zielobjekts in den Amazon S3 S3-Bucket hochgeladen. AWS SCT verwendet den Amazon S3 S3-Bucket, den Sie im Voraussetzungen Schritt im AWS Serviceprofil gespeichert haben.

```
SaveOnS3
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

Der `ConfigureStateMachine` Befehl konfiguriert Zustandsmaschinen. Dieser Befehl verwendet bis zu sechs Parameter. Stellen Sie sicher, dass Sie den Zielbereich mit einem der ersten drei Parameter aus der folgenden Liste definieren.

- `filterName`— der Name des Filters für Ihre Zielobjekte. Sie können einen Filter mit dem `CreateFilter` Befehl erstellen.
- `treePath`— der explizite Pfad zu Ihren Zielobjekten.
- `namePath`— der explizite Pfad zu einem bestimmten Zielobjekt.
- `iamRole`— der Amazon-Ressourcenname (ARN) der IAM-Rolle, die den Zugriff auf Ihre Step Machines ermöglicht. Dieser Parameter muss angegeben werden.

Im folgenden Codebeispiel werden Zustandsmaschinen konfiguriert, die `AWS_STEP_FUNCTIONS` unter Verwendung der *`role_name`* IAM-Rolle definiert sind.

```
ConfigureStateMachine
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
    -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

Der `ApplyToTarget` Befehl wendet Ihren konvertierten Code auf den Zielserver an. Um diesen Befehl auszuführen, verwenden Sie einen der folgenden Parameter:`filterName`,, oder`treePath`, `namePath` um die anzuwendenden Zielobjekte zu definieren.

Das folgende Codebeispiel wendet die `app_wp` Zustandsmaschine auf an AWS Step Functions.

```
ApplyToTarget
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Um sicherzustellen, dass Ihr konvertierter Code dieselben Ergebnisse liefert wie Ihr Quellcode, können Sie das AWS SCT Erweiterungspaket verwenden. Dies ist eine Reihe von AWS Lambda Funktionen, die Ihre Apache Oozie-Funktionen emulieren, die AWS Step Functions nicht unterstützt werden. Um dieses Erweiterungspaket zu installieren, können Sie den `CreateLambdaExtPack` Befehl verwenden.

Dieser Befehl verwendet bis zu fünf Parameter. Stellen Sie sicher, dass Sie **`Oozie2SF`** für verwende`extPackId`. AWS SCT Erstellt in diesem Fall ein Erweiterungspaket für Apache Oozie-Quellfunktionen.

- `extPackId`— der eindeutige Bezeichner für eine Reihe von Lambda-Funktionen. Dieser Parameter muss angegeben werden.
- `tempDirectory`— der Pfad, in dem temporäre Dateien gespeichert werden AWS SCT können. Dieser Parameter muss angegeben werden.
- `awsProfile`— der Name Ihres AWS Profils.
- `lambdaExecRoles`— die Liste der Amazon-Ressourcennamen (ARNs) der Ausführungsrollen, die für Lambda-Funktionen verwendet werden sollen.
- `createInvokeRoleFlag`— das boolesche Flag, das angibt, ob eine Ausführungsrolle für erstellt werden soll. AWS Step Functions

Um das Erweiterungspaket zu installieren und zu verwenden, stellen Sie sicher, dass Sie die erforderlichen Berechtigungen bereitstellen. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von AWS Lambda Funktionen im Erweiterungspaket.

Weitere Informationen zu den `CreateLambdaExtPack` Befehlen `ConvertSaveOnS3`,`ConfigureStateMachine`,`ApplyToTarget`, und finden Sie in der AWS Schema Conversion Tool CLI-Referenz.

Ausführen Ihres CLI-Skripts

Wenn Sie mit der Bearbeitung Ihres AWS SCT CLI-Skripts fertig sind, speichern Sie es als Datei mit der `.scts` Erweiterung. Jetzt können Sie Ihr Skript im `app` Ordner Ihres AWS SCT Installationspfads ausführen. Führen Sie dazu den folgenden Befehl aus.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Ersetzen Sie im vorherigen Beispiel durch *`script_path`* den Pfad zu Ihrer Datei mit dem CLI-Skript. Weitere Hinweise zum Ausführen von CLI-Skripts in AWS SCT finden Sie unterSkriptmodus.

Apache Oozie-Knoten, die konvertiert AWS SCT werden können in AWS Step Functions

Sie können AWS SCT es verwenden, um Apache Oozie-Aktionsknoten und Kontrollflussknoten in zu konvertieren. AWS Step Functions

Zu den unterstützten Aktionsknoten gehören die folgenden:

- Hive-Aktion
- Hive 2-Aktion
- Aktion auslösen
- MapReduce Streaming-Aktion
- Java-Aktion
- DistCp Aktion
- Aktion Schwein
- Sqoop-Aktion
- FS-Aktion
- Shell-Aktion

Zu den unterstützten Kontrollflussknoten gehören:

- Aktion starten
- Aktion beenden
- Aktion beenden

- Entscheidung, Aktion
- Aktion Gabel
- Mach mit bei der Aktion

Integrieren AWS Database Migration Service mit AWS Schema Conversion Tool

Verwendung eines AWS SCT Replikationsagenten mit AWS DMS

Bei sehr großen Datenbankmigrationen können Sie einen AWS SCT Replikationsagenten (aws-schema-conversion-tool-dms-agent) verwenden, um Daten aus Ihrer lokalen Datenbank auf Amazon S3 oder ein Edge-Gerät zu kopieren. AWS Snowball Edge Der Replikationsagent arbeitet zusammen mit AWS DMS und kann im Hintergrund arbeiten, während er geschlossen ist. AWS SCT

Bei der Arbeit mit AWS Snowball Edge Edge repliziert der AWS SCT Agent Daten auf das AWS Snowball Edge Gerät. Das Gerät wird dann an einen Amazon S3 S3-Bucket gesendet AWS und die Daten werden in diesen geladen. Während dieser Zeit läuft der AWS SCT Agent weiter. Der Agent nimmt dann die Daten aus Amazon S3 und kopiert diese in den Zielendpunkt.

Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.

Verwenden eines AWS SCT Datenextraktionsagenten mit AWS DMS

In AWS SCT finden Sie einen Datenextraktionsagenten (`aws-schema-conversion-tool-extractor`), der Ihnen hilft, Migrationen von Apache Cassandra zu Amazon DynamoDB zu vereinfachen. Cassandra und DynamoDB sind NoSQL-Datenbanken, unterscheiden sich jedoch in der Systemarchitektur und Datendarstellung. Sie können assistentengestützte Workflows verwenden, um den Migrationsprozess zu automatisieren. AWS SCT Cassandra-to-DynamoDB AWS SCT integriert sich in AWS Database Migration Service (AWS DMS), um die eigentliche Migration durchzuführen.

Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.

Erhöhung der Protokollierungsstufen AWS SCT bei Verwendung von AWS DMS

Sie können die Protokollierungsstufen erhöhen AWS DMS, wenn Sie AWS SCT mit arbeiten, z. B. wenn Sie mit AWS Support zusammenarbeiten müssen.

Öffnen Sie nach der Installation AWS SCT und den erforderlichen Treibern die Anwendung, indem Sie AWS SCT auf das Symbol klicken. Wenn Sie eine Aktualisierungsbenachrichtigung sehen, können Sie wählen, ob Sie das Update vor oder nach Abschluss Ihres Projekts durchführen möchten. Wenn ein Autoprojekt-Fenster geöffnet wird, schließen Sie das Fenster und erstellen Sie manuell ein Projekt.

Um die Protokollierungsstufen bei der Verwendung von AWS SCT zu erhöhen AWS DMS

1. Wählen Sie im Menü Einstellungen die Option Allgemeine Einstellungen aus.
2. Wählen Sie im Fenster Allgemeine Einstellungen die Option Protokollierung aus.
3. Wählen Sie für den Debug-Modus True.
4. Im Abschnitt Nachrichtenebene können Sie die folgenden Protokolltypen ändern:

   - Allgemeines
   - Loader
   - Parser
   - Drucker
   - Resolver
   - Telemetrie
   - Konverter

   Standardmäßig sind alle Nachrichtenstufen auf Info eingestellt.
5. Wählen Sie eine Protokollierungsebene für alle Typen von Nachrichtenebenen, die Sie ändern möchten:

   - Trace (detaillierteste Protokollierung)
   - Debuggen
   - Informationen
   - Warnung

- Fehler (am wenigsten detaillierte Protokollierung)
- Kritisch
- zwingend erforderlich

6. Wählen Sie Anwenden, um die Einstellungen für Ihr Projekt zu ändern.
7. Wählen Sie OK, um das Fenster Allgemeine Einstellungen zu schließen.

Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool

Sie können einen AWS SCT Agenten verwenden, um Daten aus Ihrem lokalen Data Warehouse zu extrahieren und zu Amazon Redshift zu migrieren. Der Agent extrahiert Ihre Daten und lädt die Daten entweder auf Amazon S3 oder, bei groß angelegten Migrationen, auf ein AWS Snowball Edge Edge-Gerät hoch. Anschließend können Sie einen AWS SCT Agenten verwenden, um die Daten nach Amazon Redshift zu kopieren.

Alternativ können Sie AWS Database Migration Service (AWS DMS) verwenden, um Daten zu Amazon Redshift zu migrieren. Der Vorteil von AWS DMS ist die Unterstützung der laufenden Replikation (Erfassung von Änderungsdaten). Um die Geschwindigkeit der Datenmigration zu erhöhen, sollten Sie jedoch mehrere AWS SCT Agenten parallel verwenden. Unseren Tests zufolge migrieren AWS SCT Agenten Daten schneller als AWS DMS um 15 bis 35 Prozent. Der Geschwindigkeitsunterschied ist auf die Datenkomprimierung, die Unterstützung der parallel Migration von Tabellenpartitionen und unterschiedliche Konfigurationseinstellungen zurückzuführen. Weitere Informationen finden Sie unter Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service.

Amazon S3 ist ein Speicher- und Abrufservice. Um Objekte in Amazon S3 zu speichern, laden Sie die Datei hoch, die Sie in einem Amazon S3-Bucket speichern möchten. Beim Hochladen der Datei können Sie Berechtigungen für das Objekt sowie beliebige Metadaten festlegen.

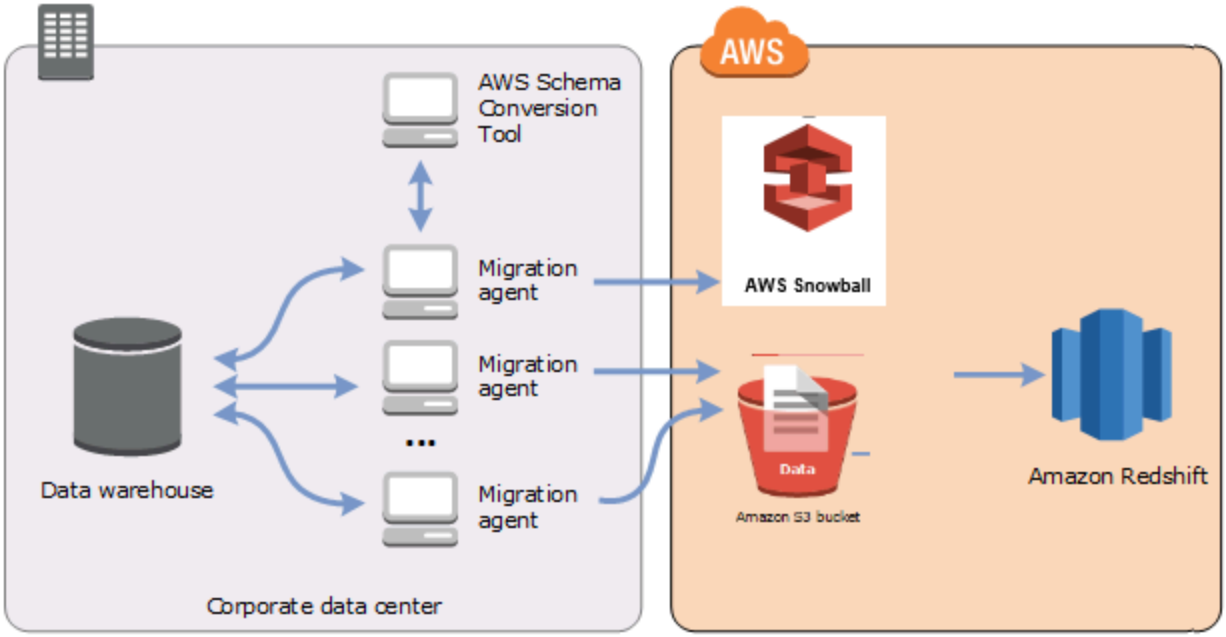
Large-scale migrations (Umfangreiche Migrationen)

Umfangreiche Datenmigrationen können viele Terabyte an Informationen umfassen und können durch die Netzwerkleistung und die schiere Menge an Daten, die verschoben werden müssen, gebremst werden. AWS Snowball Edge Edge ist ein AWS Dienst, mit dem Sie Daten mithilfe einer eigenen Appliance mit faster-than-network hoher Geschwindigkeit in die Cloud übertragen können. AWS Ein AWS Snowball Edge Edge-Gerät kann bis zu 100 TB an Daten aufnehmen. Es verwendet eine 256-Bit-Verschlüsselung und ein branchenübliches Trusted Platform Module (TPM), um sowohl die Sicherheit als auch die Vollständigkeit Ihrer Daten zu gewährleisten. chain-of-custody AWS SCT funktioniert mit Edge-Geräten. AWS Snowball Edge

Wenn Sie ein AWS Snowball Edge Edge-Gerät verwenden AWS SCT , migrieren Sie Ihre Daten in zwei Schritten. Zunächst verarbeiten Sie AWS SCT die Daten lokal und verschieben diese Daten dann auf das AWS Snowball Edge Edge-Gerät. Anschließend senden Sie das Gerät AWS mithilfe

des AWS Snowball Edge Edge-Prozesses an und laden die Daten dann AWS automatisch in einen Amazon S3 S3-Bucket. Wenn die Daten auf Amazon S3 verfügbar sind, migrieren Sie AWS SCT sie anschließend zu Amazon Redshift. Datenextraktionsagenten können im Hintergrund arbeiten, während der Vorgang geschlossen AWS SCT ist.

Das folgende Diagramm veranschaulicht das unterstützte Szenario.



Agenten für die Datenextraktion werden derzeit für die folgenden Quell-Data Warehouses unterstützt:

- Azure Synapse Analytics
- BigQuery
- Greenplum-Datenbank (Version 4.3)
- Microsoft SQL Server (Version 2008 und höher)
- Netezza (Version 7.0.3 und höher)
- Oracle (Version 10 und höher)
- Schneeflocke (Version 3)
- Teradata (Version 13 und höher)
- Vertica (Version 7.2.2 und höher)

Sie können eine Verbindung zu FIPS-Endpunkten für Amazon Redshift herstellen, wenn Sie die Sicherheitsanforderungen des Federal Information Processing Standard (FIPS) erfüllen müssen. FIPS-Endpunkte sind in den folgenden Regionen verfügbar: AWS

- Region USA Ost (Nord-Virginia) (redshift-fips.us-east-1.amazonaws.com)
- Region USA Ost (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Region USA West (Nordkalifornien) (redshift-fips.us-west-1.amazonaws.com)
- Region USA West (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Verwenden Sie die Informationen in den folgenden Themen, um zu erfahren, wie Sie mit Agenten für die Datenextraktion arbeiten.

Themen

- Voraussetzungen für die Verwendung von Datenextraktionsagenten
- Installation von Extraktionsagenten
- Konfiguration von Extraktionsagenten
- Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool
- Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen
- Regeln für die Datenmigration erstellen in AWS SCT
- Ändern der Extraktor- und Kopiereinstellungen aus den Projekteinstellungen
- Sortieren von Daten vor der Migration mit AWS SCT
- Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen
- Exportieren und Importieren einer AWS SCT Datenextraktionsaufgabe
- Datenextraktion mit einem AWS Snowball Edge Edge-Gerät
- Ausgabe der Aufgabe zur Datenextraktion
- Verwenden Sie virtuelle Partitionierung mit AWS Schema Conversion Tool
- Native Partitionierung verwenden
- Migration LOBs zu Amazon Redshift
- Bewährte Methoden und Problembehebung für Datenextraktionsagenten

Voraussetzungen für die Verwendung von Datenextraktionsagenten

Bevor Sie mit Datenextraktionsagenten arbeiten, fügen Sie Ihrem Amazon Redshift-Benutzer die erforderlichen Berechtigungen für Amazon Redshift als Ziel hinzu. Weitere Informationen finden Sie unter Berechtigungen für Amazon Redshift als Ziel.

Speichern Sie anschließend Ihre Amazon S3 S3-Bucket-Informationen und richten Sie Ihren Secure Sockets Layer (SSL) Trust and Key Store ein.

Amazon S3 S3-Einstellungen

Nachdem Ihre Agenten Ihre Daten extrahiert haben, laden sie sie in Ihren Amazon S3 S3-Bucket hoch. Bevor Sie fortfahren, müssen Sie die Anmeldeinformationen angeben, um eine Verbindung zu Ihrem AWS Konto und Ihrem Amazon S3 S3-Bucket herzustellen. Sie speichern Ihre Anmeldeinformationen und Bucket-Informationen in einem Profil in den globalen Anwendungseinstellungen und verknüpfen das Profil dann mit Ihrem AWS SCT Projekt. Wählen Sie bei Bedarf Allgemeine Einstellungen aus, um ein neues Profil zu erstellen. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.

Um Daten in Ihre Amazon Redshift Redshift-Zieldatenbank zu migrieren, benötigt der AWS SCT Datenextraktionsagent die Erlaubnis, in Ihrem Namen auf den Amazon S3 S3-Bucket zuzugreifen. Um diese Berechtigung zu erteilen, erstellen Sie einen AWS Identity and Access Management (IAM-) Benutzer mit der folgenden Richtlinie.

```
{
    "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"
        }
    ]
}
```

Ersetzen Sie es im vorherigen Beispiel *bucket_name* durch den Namen Ihres Amazon S3 S3-Buckets. Ersetzen Sie es dann *111122223333:user/DataExtractionAgentName* durch den Namen Ihres IAM-Benutzers.

IAM-Rollen annehmen

Für zusätzliche Sicherheit können Sie AWS Identity and Access Management (IAM) -Rollen für den Zugriff auf Ihren Amazon S3 S3-Bucket verwenden. Erstellen Sie dazu einen IAM-Benutzer für Ihre Datenextraktionsagenten ohne jegliche Berechtigungen. Erstellen Sie anschließend eine IAM-Rolle, die den Zugriff auf Amazon S3 ermöglicht, und geben Sie die Liste der Dienste und Benutzer an, die diese Rolle übernehmen können. Weitere Informationen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.

So konfigurieren Sie IAM-Rollen für den Zugriff auf Ihren Amazon S3 S3-Bucket

1. Erstellen Sie einen neuen IAM-Benutzer. Wählen Sie für Benutzeranmeldedaten den Zugriffstyp Programmatic aus.
2. Konfigurieren Sie die Hostumgebung so, dass Ihr Datenextraktionsagent die dafür vorgesehene Rolle übernehmen kann AWS SCT . Stellen Sie sicher, dass der Benutzer, den Sie im vorherigen Schritt konfiguriert haben, den Datenextraktionsagenten die Nutzung der Anmeldeinformationsanbieterkette ermöglicht. Weitere Informationen finden Sie unter Verwenden von Anmeldeinformationen im AWS SDK für Java Entwicklerhandbuch.
3. Erstellen Sie eine neue IAM-Rolle, die Zugriff auf Ihren Amazon S3 S3-Bucket hat.
4. Ändern Sie den Vertrauensbereich dieser Rolle, sodass der Benutzer, den Sie zuvor erstellt haben, die Rolle als vertrauenswürdig eingestuft wird. Im folgenden Beispiel *111122223333:user/DataExtractionAgentName* ersetzen Sie ihn durch den Namen Ihres Benutzers.

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
    },
    "Action": "sts:AssumeRole"
}
```

5. Ändern Sie den Vertrauensbereich dieser Rolle in Vertrauen`redshift.amazonaws.com`, um die Rolle zu übernehmen.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "redshift.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole"
}
```

6. Ordnen Sie diese Rolle Ihrem Amazon Redshift Redshift-Cluster zu.

Jetzt können Sie Ihren Datenextraktionsagenten in AWS SCT ausführen.

Wenn Sie IAM-Rollenannahme verwenden, funktioniert die Datenmigration wie folgt. Der Datenextraktionsagent wird gestartet und ruft die Benutzeranmeldeinformationen mithilfe der Anmeldeinformationsanbieterkette ab. Als Nächstes erstellen Sie eine Datenmigrationsaufgabe in AWS SCT, geben dann die IAM-Rolle an, die die Datenextraktionsagenten übernehmen sollen, und starten die Aufgabe. AWS Security Token Service (AWS STS) generiert temporäre Anmeldeinformationen für den Zugriff auf Amazon S3. Der Datenextraktionsagent verwendet diese Anmeldeinformationen, um Daten auf Amazon S3 hochzuladen.

AWS SCT Stellt dann Amazon Redshift die IAM-Rolle zur Verfügung. Im Gegenzug erhält Amazon Redshift neue temporäre Anmeldeinformationen für AWS STS den Zugriff auf Amazon S3. Amazon Redshift verwendet diese Anmeldeinformationen, um Daten von Amazon S3 in Ihre Amazon Redshift Redshift-Tabelle zu kopieren.

Sicherheitseinstellungen

Die AWS Schema Conversion Tool und die Extraktionsagenten können über Secure Sockets Layer (SSL) kommunizieren. Richten Sie zum Aktivieren von SSL einen Vertrauensspeicher und einen Schlüsselspeicher ein.

So richten Sie die sichere Kommunikation mit dem Agenten für die Datenextraktion ein

1. Starten Sie den AWS Schema Conversion Tool.
2. Öffnen Sie das Einstellungsmenü und wählen Sie dann Allgemeine Einstellungen. Das Dialogfeld Global settings wird angezeigt.
3. Wählen Sie die Option Security (Sicherheit) aus.
4. Wählen Sie Vertrauens- und Schlüsselspeicher generieren oder Bestehenden Vertrauensspeicher auswählen.

   Wenn Sie Vertrauens- und Schlüsselspeicher generieren wählen, geben Sie anschließend den Namen und das Passwort für den Vertrauens- und Schlüsselspeicher sowie den Pfad zum Speicherort für die generierten Dateien an. Diese Dateien werden in späteren Schritten verwendet.

   Wenn Sie Bestehenden Vertrauensspeicher auswählen wählen, geben Sie anschließend das Passwort und den Dateinamen für die Vertrauens- und Schlüsselspeicher an. Diese Dateien werden in späteren Schritten verwendet.
5. Nachdem Sie den Vertrauensspeicher und den Schlüsselspeicher angegeben haben, wählen Sie OK, um das Dialogfeld Allgemeine Einstellungen zu schließen.

Konfiguration der Umgebung für Datenextraktionsagenten

Sie können mehrere Datenextraktionsagenten auf einem einzigen Host installieren. Wir empfehlen jedoch, einen Datenextraktionsagenten auf einem Host auszuführen.

Um Ihren Datenextraktionsagenten auszuführen, stellen Sie sicher, dass Sie einen Host mit mindestens vier V CPUs und 32 GB Arbeitsspeicher verwenden. Stellen Sie außerdem den verfügbaren Mindestspeicher AWS SCT auf mindestens vier GB ein. Weitere Informationen finden Sie unter Konfiguration von zusätzlichem Speicher.

Die optimale Konfiguration und Anzahl der Agent-Hosts hängen von der spezifischen Situation jedes Kunden ab. Achten Sie darauf, Faktoren wie die Menge der zu migrierenden Daten, die Netzwerkbandbreite, die Zeit für das Extrahieren von Daten usw. zu berücksichtigen. Sie können zuerst einen Machbarkeitsnachweis (PoC) durchführen und dann Ihre Datenextraktionsagenten und Hosts entsprechend den Ergebnissen dieses PoC konfigurieren.

Installation von Extraktionsagenten

Wir empfehlen, mehrere Agenten für die Datenextraktion auf einzelnen Computern zu installieren. Dabei sollte auf dem Computer, auf dem das AWS Schema Conversion Tool ausgeführt wird, kein Agent installiert werden.

Agenten für die Datenextraktion werden derzeit auf folgenden Betriebssystemen installiert:

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (Version 14.04 und höher)

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu installieren. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installieren möchten.

So installieren Sie einen Agenten für die Datenextraktion

1. Wenn Sie die AWS SCT Installationsdatei noch nicht heruntergeladen haben, folgen Sie den Anweisungen unter, Installation und Konfiguration AWS Schema Conversion Tool um sie herunterzuladen. Die ZIP-Datei, die die Installationsdatei enthält, enthält auch die AWS SCT Installationsdatei für den Extraktionsagenten.

2. Laden Sie die neueste Version von Amazon Corretto 11 herunter und installieren Sie sie. Weitere Informationen finden Sie unter Downloads für Amazon Corretto 11 im Amazon Corretto 11-Benutzerhandbuch.
3. Die Installationsdatei für den Agenten für die Datenextraktion befindet sich im Unterordner "Agents". Nachfolgend ist für jedes Betriebssystem die richtige Datei für die Installation des Agenten für die Datenextraktion gezeigt.

Betriebssystem	Dateiname
Microsoft Windows	`aws-schema-conversion-tool-extractor -2.0.1.` *`build-number`* `.msi`
RHEL	`aws-schema-conversion-tool-extractor -2.0.1.` *`build-number`* `.x86_64.rpm`
Ubuntu Linux	`aws-schema-conversion-tool-extractor -2.0.1.` *`build-number`* `.deb`

4. Installieren Sie den Extraktionsagenten auf einem separaten Computer, indem Sie die Installationsdatei auf den neuen Computer kopieren.
5. Führen Sie die Installationsdatei aus. Befolgen Sie die Anleitung für Ihr Betriebssystem:

Betriebssystem	Installationsanleitungen
Microsoft Windows	Doppelklicken Sie auf die Datei, um das Installationsprogramm auszuführen.

RHEL	Führen Sie die folgenden Befehle in dem Ordner aus, in den Sie die Datei heruntergeladen oder verschoben haben.
	`sudo rpm -ivh aws-schema-conversion-tool-extractor -2.0.1.` *`build-number`* `.x86_64.rpm`
	`sudo ./sct-extractor-setup.sh --config`

Ubuntu Linux	Führen Sie die folgenden Befehle in dem Ordner aus, in den Sie die Datei heruntergeladen oder verschoben haben.

Betriebssystem	Installationsanleitungen
	`sudo dpkg -i aws-schema-conversion-tool-extractor -2.0.1. build-number .deb` `sudo ./sct-extractor-setup.sh --config`

6. Wählen Sie Weiter, akzeptieren Sie die Lizenzvereinbarung und wählen Sie Weiter.
7. Geben Sie den Pfad zur Installation des AWS SCT Datenextraktionsagenten ein und wählen Sie Weiter.
8. Wählen Sie Installieren, um Ihren Datenextraktionsagenten zu installieren.

   AWS SCT installiert Ihren Datenextraktionsagenten. Um die Installation abzuschließen, konfigurieren Sie Ihren Datenextraktionsagenten. AWS SCT startet automatisch das Konfigurations-Setup-Programm. Weitere Informationen finden Sie unter Konfiguration von Extraktionsagenten.
9. Wählen Sie Fertig stellen, um den Installationsassistenten zu schließen, nachdem Sie Ihren Datenextraktionsagenten konfiguriert haben.

Konfiguration von Extraktionsagenten

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu konfigurieren. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installiert haben.

So konfigurieren Sie einen Agenten für die Datenextraktion

1. Starten Sie das Konfigurations-Setup-Programm:

   - AWS SCT Startet das Konfigurations-Setup-Programm unter Windows automatisch während der Installation eines Datenextraktionsagenten.

     Bei Bedarf können Sie das Setup-Programm manuell starten. Führen Sie dazu die `ConfigAgent.bat` Datei in Windows aus. Sie finden diese Datei in dem Ordner, in dem Sie den Agenten installiert haben.
   - Führen Sie die `sct-extractor-setup.sh` Datei in RHEL und Ubuntu von dem Ort aus, an dem Sie den Agenten installiert haben.

Das Installationsprogramm fordert Sie zur Eingabe von Informationen auf. Es ist jeweils ein Standardwert vorkonfiguriert.

2. Akzeptieren Sie bei jeder Aufforderung den Standardwert, oder geben Sie einen neuen Wert ein.

   Geben Sie folgende Informationen an:

   - Geben Sie für Listening-Port die Portnummer ein, auf der der Agent lauscht.
   - Geben Sie für Add a source vendor den Wert yes ein und geben Sie dann Ihre Data Warehouse-Quellplattform ein.
   - Geben Sie für JDBC-Treiber den Speicherort ein, an dem Sie die JDBC-Treiber installiert haben.
   - Geben Sie unter Arbeitsordner den Pfad ein, in dem der AWS SCT Datenextraktionsagent die extrahierten Daten speichern soll. Das Arbeitsverzeichnis kann sich auf einem anderen Computer als der Agent befinden. Ein einzelnes Arbeitsverzeichnis kann für mehrere Agenten auf verschiedenen Computern verwendet werden.
   - Geben Sie für SSL-Kommunikation aktivieren den Wert yes ein.
   - Geben Sie für Schlüsselspeicher den Speicherort der Schlüsselspeicherdatei ein.
   - Geben Sie unter Schlüsselspeicher-Passwort das Passwort für den Schlüsselspeicher ein.
   - Geben Sie für Client-SSL-Authentifizierung aktivieren den Wert yes ein.
   - Geben Sie für Trust Store den Speicherort der Trust Store-Datei ein.
   - Geben Sie unter Trust Store-Passwort das Passwort für den Trust Store ein.

Das Installationsprogramm aktualisiert die Einstellungsdatei für den Agenten für die Datenextraktion. Die Einstellungsdatei heißt `settings.properties` und befindet sich im Installationsverzeichnis des Agenten für die Datenextraktion.

Nachfolgend sehen Sie eine Beispieleinstellungsdatei.

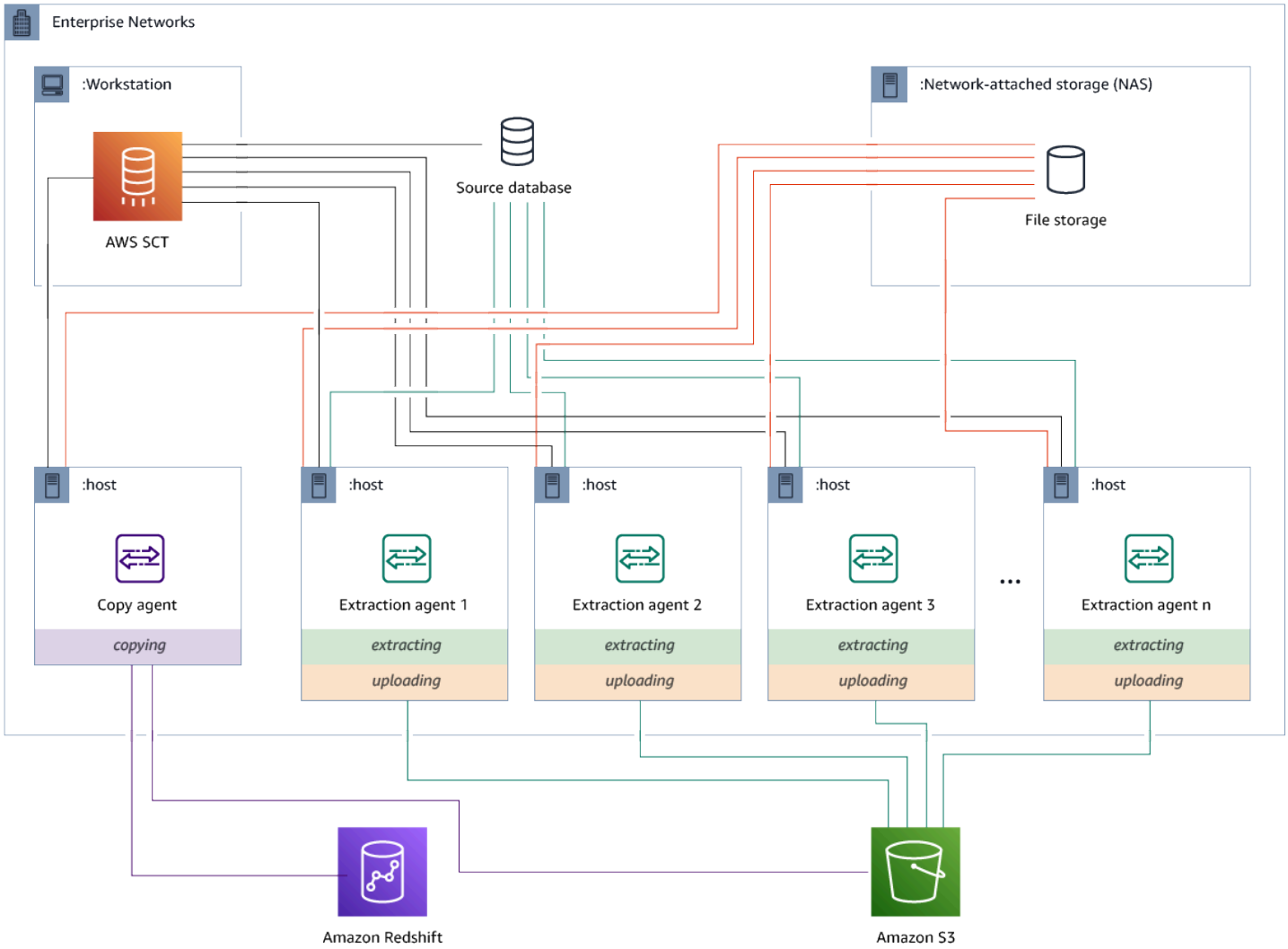
```
$ cat settings.properties
#extractor.start.fetch.size=20000
#extractor.out.file.size=10485760
#extractor.source.connection.pool.size=20
#extractor.source.connection.pool.min.evictable.idle.time.millis=30000
#extractor.extracting.thread.pool.size=10
vendor=TERADATA
```

```
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
```

Um die Konfigurationseinstellungen zu ändern, können Sie die `settings.properties` Datei mit einem Texteditor bearbeiten oder die Agentenkonfiguration erneut ausführen.

Installation und Konfiguration von Extraktionsagenten mit speziellen Kopieragenten

Sie können Extraktionsagenten in einer Konfiguration mit gemeinsam genutztem Speicher und einem dedizierten Kopieragenten installieren. Das folgende Diagramm veranschaulicht dieses Szenario.



Diese Konfiguration kann nützlich sein, wenn ein Quelldatenbankserver bis zu 120 Verbindungen unterstützt und Ihr Netzwerk über ausreichend Speicherplatz verfügt. Gehen Sie wie folgt vor, um Extraktionsagenten zu konfigurieren, die über einen eigenen Kopieragent verfügen.

So installieren und konfigurieren Sie Extraktionsagenten und einen speziellen Kopieragent

1. Stellen Sie sicher, dass das Arbeitsverzeichnis aller Extraktionsagenten denselben Ordner auf dem gemeinsam genutzten Speicher verwendet.
2. Installieren Sie die Extraktor-Agenten, indem Sie die Schritte unter befolgen. Installation von Extraktionsagenten
3. Konfigurieren Sie die Extraktionsagenten, indem Sie die Schritte unter befolgenKonfiguration von Extraktionsagenten, aber geben Sie nur den Quell-JDBC-Treiber an.

4. Konfigurieren Sie einen dedizierten Kopieragenten, indem Sie die Schritte unter Konfiguration von Extraktionsagenten befolgen, aber nur einen Amazon Redshift JDBC-Treiber angeben.

Extraktionsagenten werden gestartet

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu starten. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installiert haben.

Agenten für die Datenextraktion sind Listener. Wenn Sie einen Agenten auf diese Weise starten, wartet er auf Anweisungen. Sie senden dem Agenten später Anweisungen, Daten aus Ihrem Data Warehouse zu extrahieren.

So starten Sie einen Agenten für die Datenextraktion

- Führen Sie auf dem Computer, auf dem der Agent für die Datenextraktion installiert ist, den Befehl für Ihr Betriebssystem wie in der folgenden Tabelle dargestellt aus:

Betriebssystem	Befehl starten
Microsoft Windows	Doppelklicken Sie auf die Batch-Datei `StartAgent.bat` .
RHEL	Führen Sie in dem Verzeichnispfad, in dem Sie den Agenten installiert haben, den folgenden Befehl aus: `sudo initctl start sct-extractor`
Ubuntu Linux	Führen Sie in dem Verzeichnispfad, in dem Sie den Agenten installiert haben, den folgenden Befehl aus. Verwenden Sie den korrekten Befehl für Ihre Ubuntu-Version. Ubuntu 14.04: `sudo initctl start sct-extractor` Ubuntu 15.04 und höher: `sudo systemctl start sct-extractor`

Führen Sie zum Prüfen des Status des Agenten denselben Befehl aus und ersetzen Sie dabei `start` durch `status`.

Führen Sie zum Beenden des Agenten denselben Befehl aus und ersetzen Sie dabei `start` durch `stop`.

Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool

Sie verwalten Ihre Extraktionsagenten mithilfe von AWS SCT. Die Agenten für die Datenextraktion sind Listener. Wenn sie Anweisungen von erhalten AWS SCT, extrahieren sie Daten aus Ihrem Data Warehouse.

Gehen Sie wie folgt vor, um Extraktionsagenten für Ihr AWS SCT Projekt zu registrieren.

So registrieren Sie einen Agenten für die Datenextraktion

1. Starten Sie AWS Schema Conversion Tool das und öffnen Sie ein Projekt.
2. Öffnen Sie das Menü Ansicht und wählen Sie dann Datenmigrationsansicht (andere). Die Registerkarte Agents wird angezeigt. Wenn Sie bereits Agenten registriert haben, AWS SCT werden diese in einem Raster oben auf der Registerkarte angezeigt.
3. Wählen Sie Register aus.

   Nachdem Sie einen Agenten für ein AWS SCT Projekt registriert haben, können Sie denselben Agenten nicht mehr für ein anderes Projekt registrieren. Wenn Sie in einem AWS SCT Projekt keinen Agenten mehr verwenden, können Sie die Registrierung für ihn aufheben. Sie können ihn dann für ein anderes Projekt registrieren.
4. Wählen Sie Redshift Data Agent und dann OK.
5. Geben Sie Ihre Informationen auf der Registerkarte Verbindung des Dialogfelds ein:

   a. Geben Sie unter Beschreibung eine Beschreibung des Agenten ein.

   b. Geben Sie als Hostname den Hostnamen oder die IP-Adresse des Computers des Agenten ein.

   c. Geben Sie für Port die Portnummer ein, die der Agent abhört.

   d. Wählen Sie Registrieren, um den Agenten bei Ihrem AWS SCT Projekt zu registrieren.
6. Wiederholen Sie die obigen Schritte, um mehrere Agenten in Ihrem AWS SCT -Projekt zu registrieren.

Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen

Ein AWS SCT Agent verschlüsselt eine beträchtliche Menge an Informationen, z. B. Kennwörter für Schlüsselspeicher von Benutzern, Datenbankkonten, AWS Kontoinformationen und ähnliche Elemente. Dazu wird eine spezielle Datei mit dem Namen `seed.dat` eingesetzt. Standardmäßig erstellt der Agent die Datei im Arbeitsordner des Benutzers, der zuerst den Agenten konfiguriert.

Da unterschiedliche Benutzer den Agenten konfigurieren und ausführen können, wird der Pfad zur Datei `seed.dat` im Parameter `{extractor.private.folder}` der Datei `settings.properties` gespeichert. Wenn der Agent gestartet wird, nutzt er diesen Pfad, um die Datei `seed.dat` zu finden, damit er auf die Schlüssel-Vertrauensspeicherdaten für die Datenbank zugreifen kann, mit der er arbeitet.

In den folgenden Fällen müssen Sie vom Agenten gespeicherte Passwörter möglicherweise wiederherstellen:

- Wenn der Benutzer die `seed.dat` Datei verliert und der Standort und der Port des AWS SCT Agenten sich nicht geändert haben.
- Wenn der Benutzer die `seed.dat` Datei verliert und sich der Standort und der Port des AWS SCT Agenten geändert haben. In diesem Fall tritt die Änderung in der Regel auf, da der Agent zu einem anderen Host oder Port migriert wurde und die Informationen in der Datei `seed.dat` nicht mehr gültig sind.

Wenn in diesen Fällen der Agent ohne SSL gestartet wurde, startet der Agent und greift auf den zuvor erstellten Agentenspeicher zu. Anschließend wechselt er in den Status Waiting for recovery (Auf Wiederherstellung warten).

Wenn der Agent in diesen Fällen jedoch mit SSL gestartet wurde, können Sie ihn nicht erneut starten. Der Grund hierfür ist, dass der Agent die Passwörter für die in der Datei `settings.properties` gespeicherten Zertifikate nicht entschlüsseln kann. Bei dieser Art des Startens schlägt der Agentenstart fehl. Im Protokoll wird eine Fehlermeldung ähnlich der folgenden angezeigt: "Der Agent kann mit aktivierten SSL-Modus nicht gestartet werden. Konfigurieren Sie den Agenten neu. Grund: Das Passwort für den Schlüsselspeicher ist nicht korrekt."

Um dieses Problem zu beheben, erstellen Sie einen neuen Agenten und konfigurieren diesen so, dass er für den Zugriff auf SSL-Zertifikate die vorhandenen Passwörter nutzt. Führen Sie dazu die folgenden Schritte aus.

Nachdem Sie dieses Verfahren ausgeführt haben, sollte der Agent ausgeführt werden und in den Status Warten auf Wiederherstellung wechseln. AWS SCT sendet automatisch die benötigten Passwörter an einen Agenten mit dem Status Warten auf Wiederherstellung. Wenn der Agent über die neuen Passwörter verfügt, startet er alle Aufgaben neu. AWS SCT Nebenbei ist keine weitere Benutzeraktion erforderlich.

So konfigurieren Sie den Agenten neu und stellen Passwörter für den Zugriff auf SSL-Zertifikate wieder her

1. Installieren Sie einen neuen AWS SCT Agenten und führen Sie die Konfiguration aus.
2. Ändern Sie die `agent.name`-Eigenschaft in der Datei `instance.properties` auf den Namen des Agenten, für den der Speicher erstellt wurde, damit der neue Agent den vorhandenen Agentenspeicher nutzt.

   Die Datei `instance.properties` ist im privaten Ordner des Agenten gespeichert, der nach der folgenden Konvention benannt wird: {*output.folder*}\dmt \{*hostName*}_{*portNumber*}\.
3. Ändern Sie den Namen von {*output.folder*} in den Namen des vorherigen Ausgabeordners des Agenten.

   Derzeit versucht er immer AWS SCT noch, auf den alten Extractor auf dem alten Host und Port zuzugreifen. Deshalb erhält das Extraktionsprogramm, auf das nicht zugegriffen werden kann, den Status FAILED. Anschließend können Sie Host und Port ändern.
4. Ändern Sie den Host und/oder Port für den alten Agenten über den Befehl zum Modifizieren, um den Anforderungs-Flow zum neuen Agenten umzuleiten.

Wann AWS SCT kann der neue Agent pingen, AWS SCT erhält vom Agenten den Status Wartet auf Wiederherstellung. AWS SCT stellt dann automatisch die Passwörter für den Agenten wieder her.

Jeder Agent, der den Agentenspeicher nutzt, aktualisiert eine spezielle Datei namens `storage.lck`, die im {*output.folder*}\{*agentName*}\storage\ gespeichert ist. Diese Datei enthält die Netzwerk-ID des Agenten und die Uhrzeit, bis zu der der Speicher gesperrt ist. Wenn der Agent den Agentenspeicher nutzt, aktualisiert er die Datei `storage.lck` und erweitert das Speicher-Lease alle 5 Minuten um 10 Minuten. Vor Lease-Ablauf kann keine andere Instance diesen Agentenspeicher nutzen.

Regeln für die Datenmigration erstellen in AWS SCT

Bevor Sie Ihre Daten mit dem extrahieren AWS Schema Conversion Tool, können Sie Filter einrichten, die die Menge der extrahierten Daten reduzieren. Sie können Datenmigrationsregeln erstellen, indem Sie `WHERE` Klauseln verwenden, um die Menge der zu extrahierenden Daten zu reduzieren. Sie können beispielsweise eine `WHERE`-Klausel schreiben, die Daten aus einer einzelnen Tabelle auswählt.

Sie können Datenmigrationsregeln erstellen und die Filter als Teil Ihres Projekts speichern. Gehen Sie bei geöffnetem Projekt wie folgt vor, um Datenmigrationsregeln zu erstellen.

Um Regeln für die Datenmigration zu erstellen

1. Öffnen Sie das Menü Ansicht und wählen Sie dann Datenmigrationsansicht (andere).
2. Wählen Sie Datenmigrationsregeln und dann Neue Regel hinzufügen aus.
3. Konfigurieren Sie Ihre Datenmigrationsregel:
    a. Geben Sie unter Name einen Namen für Ihre Datenmigrationsregel ein.
    b. Geben Sie unter Where schema name is like einen Filter ein, der auf Schemas angewendet werden soll. In diesem Filter wird eine `WHERE`-Klausel wird mithilfe einer `LIKE`-Klausel ausgewertet. Um ein Schema auszuwählen, geben Sie einen exakten Schemanamen ein. Um mehrere Schemas auszuwählen, verwenden Sie das Zeichen „%“ als Platzhalter für eine beliebige Anzahl von Zeichen im Schemanamen.
    c. Geben Sie für Tabellennamen wie einen Filter ein, der auf Tabellen angewendet werden soll. In diesem Filter wird eine `WHERE`-Klausel wird mithilfe einer `LIKE`-Klausel ausgewertet. Um eine Tabelle auszuwählen, geben Sie einen exakten Namen ein. Um mehrere Tabellen auszuwählen, verwenden Sie das Zeichen „%“ als Platzhalter für eine beliebige Anzahl von Zeichen im Tabellennamen.
    d. Geben Sie für Where-Klausel eine `WHERE` Klausel ein, um Daten zu filtern.
4. Nachdem Sie Ihren Filter konfiguriert haben, wählen Sie Save, um Ihren Filter zu speichern, oder Cancel, um die Änderungen zu verwerfen.
5. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen von Filtern fertig sind, wählen Sie Alle speichern, um alle Ihre Änderungen zu speichern.

Mit dem Umschaltsymbol lässt sich ein Filter deaktivieren, ohne ihn zu löschen. Das Kopiersymbol verwenden Sie, um einen vorhandenen Filter zu duplizieren. Mit dem Löschsymbol können Sie einen

vorhandenen Filter löschen. Um alle Änderungen, die Sie an Ihren Filtern vornehmen, zu speichern, wählen Sie Alle speichern.

Ändern der Extraktor- und Kopiereinstellungen aus den Projekteinstellungen

Im Fenster mit den Projekteinstellungen in AWS SCT können Sie Einstellungen für Datenextraktionsagenten und den Amazon Redshift `COPY` Redshift-Befehl auswählen.

Um diese Einstellungen auszuwählen, wählen Sie Einstellungen, Projekteinstellungen und dann Datenmigration. Hier können Sie die Extraktionseinstellungen, Amazon S3 S3-Einstellungen und Kopiereinstellungen bearbeiten.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Extraktionseinstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Compression format (Komprimierungsformat)	Geben Sie das Komprimierungsformat der Eingabedateien an. Wählen Sie eine der folgenden Optionen: GZIP BZIP2, STD oder Keine Komprimierung.
Trennzeichen	Geben Sie das ASCII-Zeichen an, das Felder in den Eingabeda teien voneinander trennt. Nicht druckbare Zeichen werden nicht unterstützt.
NULL-Wert als Zeichenfolge	Aktivieren Sie diese Option, wenn Ihre Daten ein Nullabsch lusszeichen enthalten. Wenn diese Option ausgeschaltet ist, behandelt der Amazon Redshift `COPY` Redshift-Befehl Null als Ende des Datensatzes und beendet den Ladevorgang.
Strategie zur Sortierung	Verwenden Sie die Sortierung, um die Extraktion an der Fehlerstelle neu zu starten. Wählen Sie eine der folgenden Sortierstrategien: Verwenden Sie die Sortierung nach dem ersten Fehler (empfohlen), Verwenden Sie die Sortierung, wenn möglich, oder Verwenden Sie die Sortierung nie. Weitere Informationen finden Sie unter the section called “Daten sortieren”.

Für diesen Parameter	Vorgehensweise
Temporäres Quellschema	Geben Sie den Namen des Schemas in die Quelldatenbank ein, in der der Extraktionsagent die temporären Objekte erstellen kann.
Größe der Ausgangsdatei (in MB)	Geben Sie die Größe der auf Amazon S3 hochgeladenen Dateien in MB ein.
Größe der Snowball-Out-Datei (in MB)	Geben Sie die Größe der hochgeladenen Dateien in MB ein. AWS Snowball Edge Dateien können eine Größe von 1—1.000 MB haben.
Verwenden Sie die automatische Partitionierung. Geben Sie für Greenplum und Netezza die Mindestgröße der unterstützten Tabellen ein (in Megabyte)	Aktivieren Sie diese Option, um die Tabellenpartitionierung zu verwenden, und geben Sie dann die Größe der zu partitionierenden Tabellen für Greenplum- und Netezza-Quelldatenbanken ein. Bei Migrationen von Oracle zu Amazon Redshift können Sie dieses Feld leer lassen, da Unteraufgaben für alle AWS SCT partitionierten Tabellen erstellt werden.
Extrahieren LOBs	Aktivieren Sie diese Option, um große Objekte (LOBs) aus Ihrer Quelldatenbank zu extrahieren. LOBs schließen Sie BLOBs CLOBs, NCLOBs,, XML-Dateien usw. ein. AWS SCT Extraktionsagenten erstellen für jedes LOB eine Datendatei.
Amazon S3 LOBs S3-Bucket-Ordner	Geben Sie den Speicherort für die AWS SCT Extraktionsagenten ein LOBs.
Wenden Sie RTRIM auf Zeichenkettenspalten an	Aktivieren Sie diese Option, um einen bestimmten Zeichensatz vom Ende der extrahierten Zeichenketten abzuschneiden.
Bewahren Sie Dateien nach dem Upload auf Amazon S3 lokal auf	Aktivieren Sie diese Option, um Dateien auf Ihrem lokalen Computer zu behalten, nachdem Datenextraktionsagenten sie auf Amazon S3 hochgeladen haben.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Amazon S3 S3-Einstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Verwenden Sie einen Proxy	Aktivieren Sie diese Option, um einen Proxy-Server zum Hochladen von Daten auf Amazon S3 zu verwenden. Wählen Sie dann das Datenübertragungsprotokoll und geben Sie den Hostnamen, den Port, den Benutzernamen und das Passwort ein.
Endpunkttyp	Wählen Sie FIPS, um den FIPS-Endpunkt (Federal Information Processing Standard) zu verwenden. Wählen Sie VPCE, um den Virtual Private Cloud (VPC) -Endpunkt zu verwenden. Geben Sie dann für VPC-Endpunkt das Domain Name System (DNS) Ihres VPC-Endpunkts ein.
Dateien nach dem Kopieren nach Amazon Redshift auf Amazon S3 behalten	Aktivieren Sie diese Option, um die extrahierten Dateien auf Amazon S3 zu behalten, nachdem Sie diese Dateien nach Amazon Redshift kopiert haben.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Kopiereinstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Maximale Anzahl von Fehlern	Geben Sie die Anzahl der Ladefehler ein. Wenn der Vorgang dieses Limit erreicht hat, beenden die AWS SCT Datenextr aktionsagenten den Datenladevorgang. Der Standardwert ist 0, was bedeutet, dass die AWS SCT Datenextraktionsagenten den Datenladevorgang unabhängig von den Fehlern fortsetzen.
Ersetzt ungültige UTF-8-Zei chen	Aktivieren Sie diese Option, um ungültige UTF-8-Zeichen durch das angegebene Zeichen zu ersetzen und den Datenlade vorgang fortzusetzen.
Verwenden Sie das Leerzeich en als Nullwert	Aktivieren Sie diese Option, um leere Felder, die aus Leerzeich en bestehen, als Null zu laden.

Für diesen Parameter	Vorgehensweise
Verwenden Sie leer als Nullwert	Aktivieren Sie diese Option, um leere VARCHAR Felder CHAR und Felder mit dem Wert Null zu laden.
Spalten kürzen	Aktivieren Sie diese Option, um Daten in Spalten so zu kürzen, dass sie der Datentypspezifikation entsprechen.
Automatische Komprimierung	Aktivieren Sie diese Option, um während eines Kopiervorgangs eine Komprimierungskodierung anzuwenden.
Automatische Aktualisierung der Statistiken	Aktivieren Sie diese Option, um die Statistiken am Ende eines Kopiervorgangs zu aktualisieren.
Überprüfen Sie die Datei vor dem Laden	Aktivieren Sie diese Option, um Datendateien zu überprüfen, bevor Sie sie in Amazon Redshift laden.

Sortieren von Daten vor der Migration mit AWS SCT

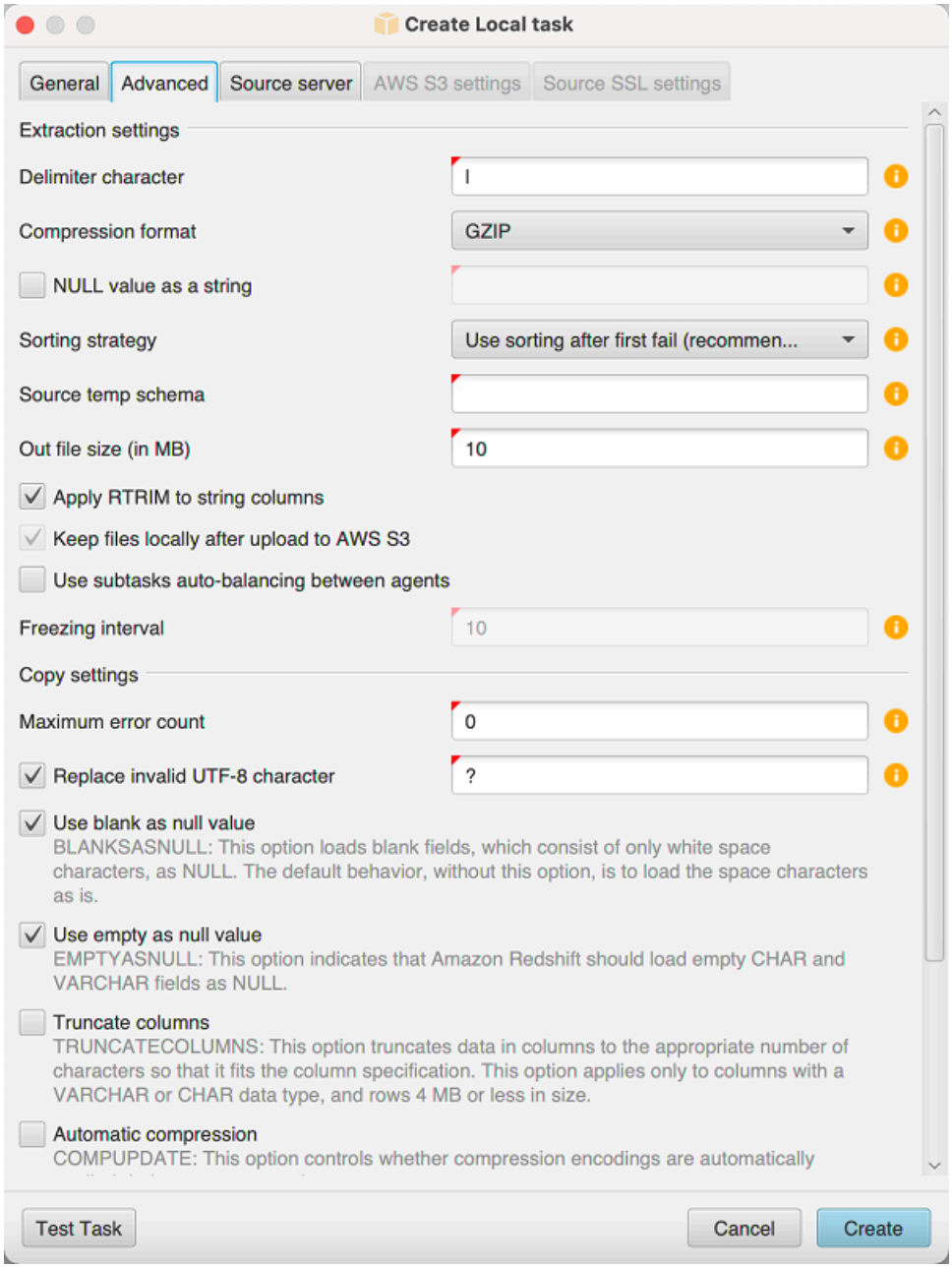
Das Sortieren Ihrer Daten vor der Migration mit AWS SCT bietet einige Vorteile. Wenn Sie die Daten zuerst sortieren, AWS SCT können Sie den Extraktionsagenten nach einem Fehler am zuletzt gespeicherten Punkt neu starten. Wenn Sie Daten zu Amazon Redshift migrieren und die Daten zuerst sortieren, AWS SCT können Sie Daten außerdem schneller in Amazon Redshift einfügen.

Diese Vorteile haben damit zu tun, wie Datenextraktionsabfragen AWS SCT erstellt werden. In einigen Fällen AWS SCT verwendet die Analysefunktion DENSE_RANK in diesen Abfragen. DENSE_RANK kann jedoch viel Zeit und Serverressourcen in Anspruch nehmen, um den Datensatz zu sortieren, der sich aus der Extraktion ergibt. Wenn also auch ohne ihn funktioniert, funktioniert er AWS SCT auch.

Um Daten vor der Migration zu sortieren, verwenden Sie AWS SCT

1. Öffnen Sie ein AWS SCT Projekt.
2. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie dann Lokale Aufgabe erstellen.
3. Wählen Sie die Registerkarte Erweitert und wählen Sie unter Sortierstrategie eine Option aus:

- Verwenden Sie niemals die Sortierung — Der Extraktionsagent verwendet die Analysefunktion DENSE_RANK nicht und startet bei einem Fehler von vorne neu.
- Verwenden Sie nach Möglichkeit die Sortierung — Der Extraktionsagent verwendet DENSE_RANK, wenn die Tabelle einen Primärschlüssel oder eine eindeutige Einschränkung hat.
- Sortierung nach dem ersten Fehler verwenden (empfohlen) — Der Extraktionsagent versucht zunächst, die Daten abzurufen, ohne DENSE_RANK zu verwenden. Schlägt der erste Versuch fehlt, baut der Agent für die Extraktion die Abfrage unter Verwendung von DENSE_RANK um und behält im Fehlerfall ihren Speicherort bei.



4. Legen Sie zusätzliche Parameter fest wie folgt, und wählen Sie Create, um Ihre Aufgabe zum Extrahieren Ihrer Daten zu erstellen.

Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen

Gehen Sie wie folgt vor, um Datenextraktionsaufgaben zu erstellen, auszuführen und zu überwachen.

So weisen Sie Agenten Aufgaben zu und migrieren Daten

1. Nachdem Sie Ihr Schema konvertiert haben, wählen Sie im linken Bereich Ihres Projekts eine oder mehrere Tabellen aus. AWS Schema Conversion Tool

   Sie können alle Tabellen auswählen. Um die Leistung nicht zu sehr zu beeinträchtigen, raten wir jedoch von dieser Vorgehensweise ab. Wir empfehlen, für mehrere Tabellen mehrere Aufgaben basierend auf der Größe der Tabellen in Ihrem Data Warehouse zu erstellen.
2. Öffnen Sie das Kontextmenü (Rechtsklick) für jede Tabelle und wählen Sie dann Aufgabe erstellen. Das Dialogfeld Lokale Aufgabe erstellen wird geöffnet.
3. Geben Sie unter Aufgabenname einen Namen für die Aufgabe ein.
4. Wählen Sie für den Migrationsmodus eine der folgenden Optionen:

   - Nur Extrahieren — Extrahieren Sie Ihre Daten und speichern Sie die Daten in Ihren lokalen Arbeitsordnern.
   - Extrahieren und hochladen — Extrahieren Sie Ihre Daten und laden Sie Ihre Daten auf Amazon S3 hoch.
   - Extrahieren, Hochladen und Kopieren — Extrahieren Sie Ihre Daten, laden Sie Ihre Daten auf Amazon S3 hoch und kopieren Sie sie in Ihr Amazon Redshift Data Warehouse.
5. Wählen Sie als Verschlüsselungstyp eine der folgenden Optionen aus:

   - KEINE — Deaktiviert die Datenverschlüsselung für den gesamten Datenmigrationsprozess.
   - CSE_SK — Verwenden Sie eine clientseitige Verschlüsselung mit einem symmetrischen Schlüssel, um Daten zu migrieren. AWS SCT generiert automatisch Verschlüsselungsschlüssel und überträgt sie mithilfe von Secure Sockets Layer (SSL) an Datenextraktionsagenten. AWS SCT verschlüsselt keine großen Objekte (LOBs) während der Datenmigration.
6. Wählen Sie Extrahieren LOBs, um große Objekte zu extrahieren. Wenn Sie keine großen Objekte extrahieren müssen, können Sie das Kontrollkästchen deaktivieren. Dadurch verringert sich die Menge der Daten, die Sie extrahieren.
7. Um detaillierte Informationen zu einer Aufgabe anzuzeigen, wählen Sie Aufgabenprotokollierung aktivieren. Sie können das Aufgabenprotokoll zum Debuggen von Problemen verwenden.

   Wenn Sie die Aufgabenprotokollierung aktivieren, wählen Sie die gewünschte Detailebene aus. Es sind folgende Ebenen verfügbar, wobei jede Ebene alle Nachrichten der darunterliegenden Ebene(n) enthält:

- ERROR— Die kleinste Menge an Details.
- WARNING
- INFO
- DEBUG
- TRACE— Die größte Menge an Details.

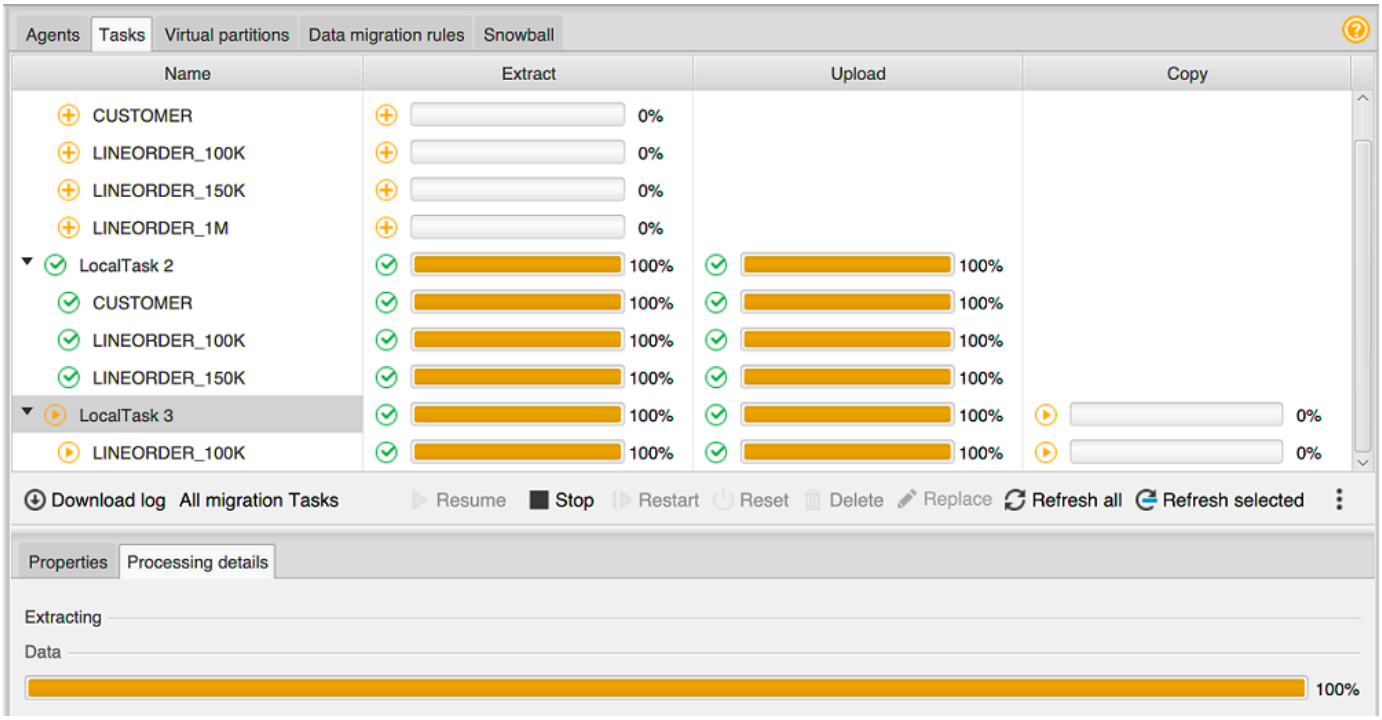
8. AWS SCT Verwendet zum Exportieren von BigQuery Daten den Bucket-Ordner von Google Cloud Storage. In diesem Ordner speichern Datenextraktionsagenten Ihre Quelldaten.

   Um den Pfad zu Ihrem Google Cloud Storage-Bucket-Ordner einzugeben, wählen Sie Erweitert. Geben Sie für den Google CS-Bucket-Ordner den Bucket-Namen und den Ordnernamen ein.

9. Um eine Rolle für den Benutzer Ihres Datenextraktionsagenten zu übernehmen, wählen Sie Amazon S3 S3-Einstellungen. Geben Sie für die IAM-Rolle den Namen der zu verwendenden Rolle ein. Wählen Sie für Region die AWS-Region für diese Rolle aus.
10. Wählen Sie Testaufgabe, um zu überprüfen, ob Sie eine Verbindung zu Ihrem Arbeitsordner, Amazon S3 S3-Bucket und Amazon Redshift Data Warehouse herstellen können. Die Überprüfung ist abhängig vom gewählten Migrationsmodus.
11. Klicken Sie auf Create, um die Aufgabe zu erstellen.
12. Wiederholen Sie die vorherigen Schritte, um Aufgaben für alle zu migrierenden Daten zu erstellen.

So können Sie Aufgaben ausführen und überwachen

1. Wählen Sie für Ansicht die Datenmigrationsansicht aus. Die Registerkarte Agents wird angezeigt.
2. Wählen Sie die Registerkarte Tasks aus. Ihre Aufgaben werden im Raster oben wie folgt angezeigt. Der Status einer Aufgabe wird im oberen Raster und der Status der Unteraufgaben im unteren Raster angezeigt.



3. Wählen Sie oben im Raster eine Aufgabe aus und erweitern Sie diese. Abhängig vom gewählten Migrationsmodus ist die Aufgabe in die Punkte Extract, Upload und Copy unterteilt.
4. Wählen Sie Start für eine Aufgabe, um sie zu starten. Sie können den Status Ihrer Aufgaben während der Ausführung überwachen. Die Unteraufgaben werden parallel ausgeführt. Auch das Extrahieren, Hochladen und Kopieren erfolgt parallel.
5. Wenn Sie beim Einrichten der Aufgabe die Protokollierung aktiviert haben, können Sie das Protokoll anzeigen:

    a. Wählen Sie Protokoll herunterladen aus. Es wird eine Nachricht mit dem Namen des Verzeichnisses angezeigt, das die Protokolldatei enthält. Schließen Sie die Nachricht.

    b. Auf der Registerkarte Task details wird ein Link angezeigt. Klicken Sie auf den Link, um das Verzeichnis zu öffnen, das die Protokolldatei enthält.

Sie können den Vorgang schließen AWS SCT und Ihre Agenten und Aufgaben werden weiterhin ausgeführt. Sie können das Programm AWS SCT später erneut öffnen, um den Status Ihrer Aufgaben zu überprüfen und die Task-Protokolle einzusehen.

Sie können Datenextraktionsaufgaben auf Ihrer lokalen Festplatte speichern und sie mithilfe von Export und Import in demselben oder einem anderen Projekt wiederherstellen. Um eine Aufgabe zu exportieren, stellen Sie sicher, dass in einem Projekt mindestens eine Extraktionsaufgabe erstellt

wurde. Sie können eine einzelne Extraktionsaufgabe oder alle im Projekt erstellten Aufgaben importieren.

Wenn Sie eine Extraktionsaufgabe exportieren, AWS SCT erstellt eine separate `.xml` Datei für diese Aufgabe. In der `.xml` Datei werden die Metadateninformationen dieser Aufgabe gespeichert, z. B. Aufgabeneigenschaften, Beschreibung und Unteraufgaben. Die `.xml` Datei enthält keine Informationen zur Verarbeitung einer Extraktionsaufgabe. Informationen wie die folgenden werden beim Import der Aufgabe neu erstellt:

- Fortschritt der Aufgabe
- Status von Unteraufgaben und Phasen
- Verteilung der Extraktionsmittel nach Unteraufgaben und Stufen
- Aufgabe und Unteraufgabe IDs
- Task name

Exportieren und Importieren einer AWS SCT Datenextraktionsaufgabe

Mithilfe von AWS SCT Export und Import können Sie schnell eine vorhandene Aufgabe aus einem Projekt speichern und sie in einem anderen Projekt (oder demselben Projekt) wiederherstellen. Gehen Sie wie folgt vor, um Datenextraktionsaufgaben zu exportieren und zu importieren.

Um eine Datenextraktionsaufgabe zu exportieren und zu importieren

1. Wählen Sie für Ansicht die Datenmigrationsansicht aus. Die Registerkarte Agents wird angezeigt.
2. Wählen Sie die Registerkarte Tasks aus. Ihre Aufgaben werden in dem sich öffnenden Raster aufgeführt.
3. Wählen Sie die drei vertikal ausgerichteten Punkte (Ellipsensymbol) in der unteren rechten Ecke unter der Aufgabenliste aus.
4. Wählen Sie im Popup-Menü die Option „Aufgabe exportieren“.
5. Wählen Sie den Ordner aus, in dem Sie die `.xml` Aufgabenexportdatei ablegen möchten AWS SCT .

   AWS SCT erstellt die Aufgabenexportdatei mit dem Dateinamenformat *`TASK-DESCRIPTION_TASK-ID`*`.xml`.

6. Wählen Sie die drei vertikal ausgerichteten Punkte (Ellipsensymbol) unten rechts unter der Aufgabenliste aus.
7. Wählen Sie im Popup-Menü die Option „Aufgabe importieren“.

   Sie können eine Extraktionsaufgabe in ein Projekt importieren, das mit der Quelldatenbank verbunden ist, und das Projekt hat mindestens einen aktiven registrierten Extraktionsagenten.
8. Wählen Sie die `.xml` Datei für die Extraktionsaufgabe aus, die Sie exportiert haben.

   AWS SCT ruft die Parameter der Extraktionsaufgabe aus der Datei ab, erstellt die Aufgabe und fügt sie den Extraktionsagenten hinzu.
9. Wiederholen Sie diese Schritte, um zusätzliche Datenextraktionsaufgaben zu exportieren und zu importieren.

Am Ende dieses Vorgangs sind Ihr Export und Import abgeschlossen und Ihre Datenextraktionsaufgaben sind einsatzbereit.

Datenextraktion mit einem AWS Snowball Edge Edge-Gerät

Die Verwendung AWS SCT von AWS Snowball Edge Edge besteht aus mehreren Schritten. Die Migration umfasst eine lokale Aufgabe, bei der ein Datenextraktionsagent AWS SCT verwendet wird, um die Daten auf das AWS Snowball Edge Edge-Gerät zu verschieben, und anschließend eine Zwischenaktion, bei der die Daten vom AWS Snowball Edge Edge-Gerät in einen Amazon S3 S3-Bucket AWS kopiert werden. Der Vorgang beendet das AWS SCT Laden der Daten aus dem Amazon S3 S3-Bucket in Amazon Redshift.

Die Abschnitte, die dieser Übersicht folgen, enthalten eine step-by-step Anleitung zu jeder dieser Aufgaben. Das Verfahren setzt voraus, dass Sie einen Datenextraktionsagenten auf einem speziellen Computer AWS SCT installiert und konfiguriert und registriert haben.

Führen Sie die folgenden Schritte aus, um Daten mithilfe von AWS Snowball Edge Edge von einem lokalen Datenspeicher zu einem AWS Datenspeicher zu migrieren.

1. Erstellen Sie mit der AWS Snowball Edge Konsole einen AWS Snowball Edge Edge-Job.
2. Entsperren Sie das AWS Snowball Edge Edge-Gerät mithilfe des lokalen, dedizierten Linux-Computers.
3. Erstellen Sie ein neues Projekt in AWS SCT.
4. Installieren und konfigurieren Sie Ihre Datenextraktionsagenten.

5. Erstellen Sie Berechtigungen für den zu verwendenden Amazon S3-Bucket und richten Sie diese ein.
6. Importieren Sie einen AWS Snowball Edge Job in Ihr AWS SCT Projekt.
7. Registrieren Sie Ihren Datenextraktionsagenten in AWS SCT.
8. Erstellen Sie eine lokale Aufgabe in AWS SCT.
9. Führen Sie die Datenmigrationsaufgabe in aus und überwachen Sie sie AWS SCT.

Step-by-step Verfahren für die Datenmigration mithilfe von AWS SCT Edge AWS Snowball Edge

Die folgenden Abschnitte enthalten ausführliche Informationen über die einzelnen Migrationsschritte.

Schritt 1: Erstellen Sie einen AWS Snowball Edge Edge-Job

Erstellen Sie einen AWS Snowball Edge Job, indem Sie die im Abschnitt Erstellen eines AWS Snowball Edge Edge-Auftrags im Edge-Entwicklerhandbuch beschriebenen Schritte ausführen.AWS Snowball Edge

Schritt 2: Entsperren Sie das AWS Snowball Edge Edge-Gerät

Führen Sie die Befehle zum Entsperren und Bereitstellen von Anmeldeinformationen für das Snowball Edge Edge-Gerät von dem Computer aus, auf dem Sie den AWS DMS Agenten installiert haben. Wenn Sie diese Befehle ausführen, können Sie sicherstellen, dass der AWS DMS Agentenanruf eine Verbindung zum AWS Snowball Edge Edge-Gerät herstellt. Weitere Informationen zum Entsperren des AWS Snowball Edge Edge-Geräts finden Sie unter Entsperren des Snowball Edge Edge.

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://
<Snowball IP>:8080 --recursive
```

Schritt 3: Erstellen Sie ein neues Projekt AWS SCT

Als Nächstes erstellen Sie ein neues AWS SCT Projekt.

Um ein neues Projekt zu erstellen in AWS SCT

1. Starte das AWS Schema Conversion Tool. Wählen Sie im Menü Datei die Option Neues Projekt aus. Das Dialogfeld Neues Projekt wird angezeigt.

2. Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.
3. Geben Sie den Speicherort für Ihre lokale Projektdatei an.
4. Wählen Sie OK, um Ihr AWS SCT Projekt zu erstellen.
5. Wählen Sie Quelle hinzufügen, um Ihrem AWS SCT Projekt eine neue Quelldatenbank hinzuzufügen.
6. Wählen Sie Ziel hinzufügen, um Ihrem AWS SCT Projekt eine neue Zielplattform hinzuzufügen.
7. Wählen Sie im linken Bereich das Quelldatenbankschema aus.
8. Geben Sie im rechten Bereich die Zieldatenbankplattform für das ausgewählte Quellschema an.
9. Wählen Sie Zuordnung erstellen. Diese Schaltfläche wird aktiv, nachdem Sie das Quelldatenbankschema und die Zieldatenbankplattform ausgewählt haben.

Schritt 4: Installieren und konfigurieren Sie Ihren Datenextraktionsagenten

AWS SCT verwendet einen Datenextraktionsagenten, um Daten zu Amazon Redshift zu migrieren. Die ZIP-Datei, die Sie zur Installation heruntergeladen haben AWS SCT, enthält die Installationsdatei für den Extraktionsagenten. Sie können den Datenextraktionsagenten unter Windows, Red Hat Enterprise Linux oder Ubuntu installieren. Weitere Informationen finden Sie unter Installation von Extraktionsagenten.

Um Ihren Datenextraktionsagenten zu konfigurieren, geben Sie Ihre Quell- und Zieldatenbank-Engines ein. Stellen Sie außerdem sicher, dass Sie die JDBC-Treiber für Ihre Quell- und Zieldatenbanken auf den Computer heruntergeladen haben, auf dem Sie Ihren Datenextraktionsagenten ausführen. Datenextraktionsagenten verwenden diese Treiber, um eine Verbindung zu Ihren Quell- und Zieldatenbanken herzustellen. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

In Windows startet das Installationsprogramm für den Datenextraktionsagenten den Konfigurationsassistenten im Befehlszeilenfenster. Unter Linux führen Sie die `sct-extractor-setup.sh` Datei von dem Ort aus, an dem Sie den Agenten installiert haben.

Schritt 5: Konfiguration AWS SCT für den Zugriff auf den Amazon S3 S3-Bucket

Informationen zur Konfiguration eines Amazon S3 S3-Buckets finden Sie unter Buckets-Übersicht im Amazon Simple Storage Service-Benutzerhandbuch.

Schritt 6: Importieren Sie einen AWS Snowball Edge Job in Ihr Projekt AWS SCT

Importieren Sie Ihren AWS Snowball Edge Job, um Ihr AWS SCT Projekt mit Ihrem AWS Snowball Edge Edge-Gerät zu verbinden.

Um deinen AWS Snowball Edge Job zu importieren

1. Öffnen Sie das Einstellungsmenü und wählen Sie dann Allgemeine Einstellungen. Das Dialogfeld Global settings wird angezeigt.
2. Wählen Sie AWS Dienstprofile und dann Job importieren aus.
3. Wählen Sie Ihren AWS Snowball Edge Job aus.
4. Gib deine AWS Snowball Edge IP ein. Weitere Informationen finden Sie im AWS Snowball Edge Benutzerhandbuch unter Ändern Ihrer IP-Adresse.
5. Geben Sie Ihren AWS Snowball Edge Port ein. Weitere Informationen finden Sie unter Für die Nutzung von AWS Diensten auf einem AWS Snowball Edge Edge-Gerät erforderliche Ports im AWS Snowball Edge Edge-Entwicklerhandbuch.
6. Geben Sie Ihren AWS Snowball Edge Zugriffsschlüssel und Ihren AWS Snowball Edge geheimen Schlüssel ein. Weitere Informationen finden Sie unter Autorisierung und Zugriffskontrolle AWS Snowball Edge im AWS Snowball Edge Benutzerhandbuch.
7. Wählen Sie Übernehmen und anschließend OK aus.

Schritt 7: Registrieren Sie einen Datenextraktionsagenten in AWS SCT

In diesem Abschnitt registrieren Sie den Datenextraktionsagenten in AWS SCT.

Um einen Datenextraktionsagenten zu registrieren

1. Wählen Sie im Menü Ansicht die Option Datenmigrationsansicht (andere) und wählen Sie dann Registrieren aus.
2. Geben Sie unter Beschreibung einen Namen für Ihren Datenextraktionsagenten ein.
3. Geben Sie als Hostname die IP-Adresse des Computers ein, auf dem Sie Ihren Datenextraktionsagenten ausführen.
4. Geben Sie unter Port den Listening-Port ein, den Sie konfiguriert haben.
5. Wählen Sie Register aus.

Schritt 8: Erstellen einer lokalen Aufgabe

Als Nächstes erstellen Sie die Migrationsaufgabe. Die Aufgabe umfasst zwei Unteraufgaben. Eine Unteraufgabe migriert Daten aus der Quelldatenbank zur AWS Snowball Edge Edge-Appliance. Die andere Unteraufgabe verschiebt die von der Appliance geladenen Daten in einen Amazon S3-Bucket und migriert sie in die Zieldatenbank.

Um die Migrationsaufgabe zu erstellen

1. Wählen Sie im Menü Ansicht die Option Datenmigrationsansicht (andere) aus.
2. Wählen Sie im linken Bereich, der das Schema Ihrer Quelldatenbank anzeigt, ein Schemaobjekt aus, das migriert werden soll. Öffnen Sie das Kontextmenü (mit der rechten Maustaste) für das Objekt und wählen Sie dann Lokale Aufgabe erstellen.
3. Geben Sie unter Aufgabenname einen aussagekräftigen Namen für Ihre Datenmigrationsaufgabe ein.
4. Wählen Sie für den Migrationsmodus Extrahieren, Hochladen und Kopieren aus.
5. Wählen Sie Amazon S3 S3-Einstellungen.
6. Wählen Sie Snowball Edge verwenden aus.
7. Geben Sie Ordner und Unterordner in Ihrem Amazon S3 S3-Bucket ein, in denen der Datenextraktionsagent Daten speichern kann.
8. Klicken Sie auf Create, um die Aufgabe zu erstellen.

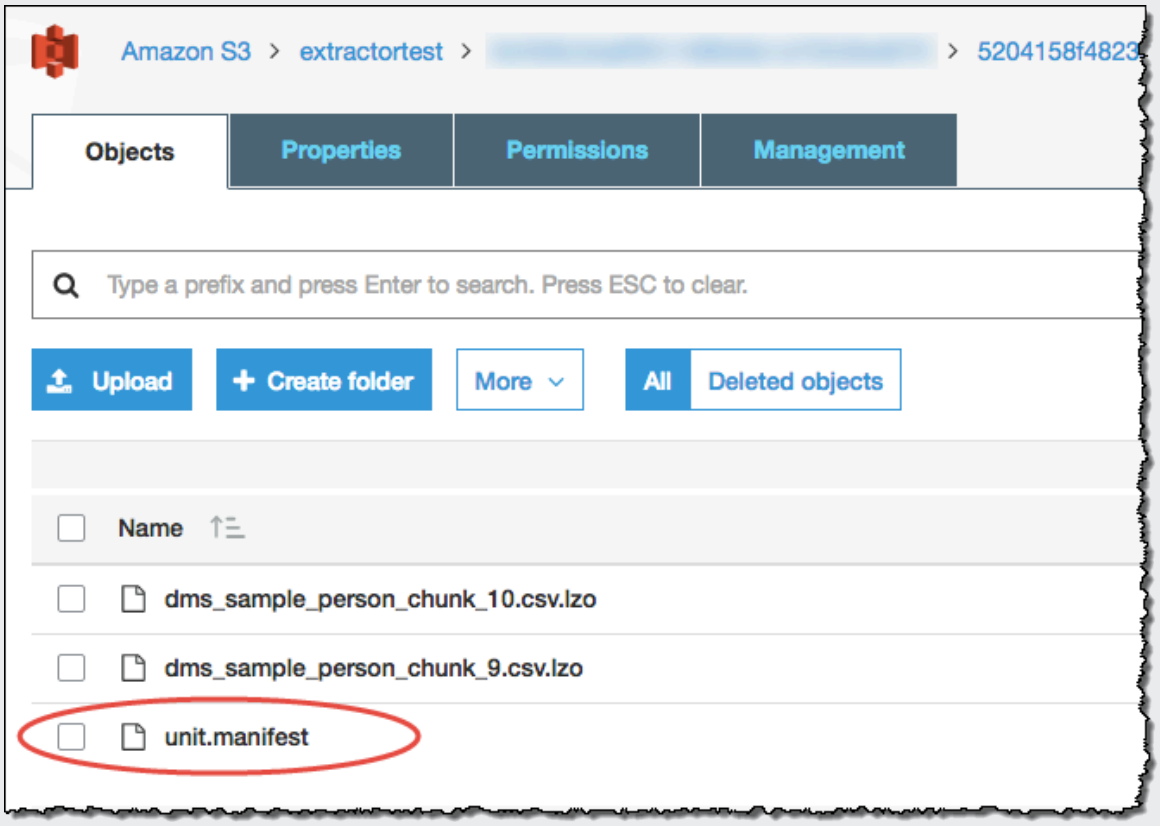
Schritt 9: Ausführen und Überwachen der Datenmigrationsaufgabe in AWS SCT

Um Ihre Datenmigrationsaufgabe zu starten, wählen Sie Start. Stellen Sie sicher, dass Sie Verbindungen zur Quelldatenbank, zum Amazon S3 S3-Bucket, zum AWS Snowball Edge Gerät sowie zur Verbindung mit der Zieldatenbank hergestellt haben AWS.

Sie können die Datenmigrationsaufgaben und ihre Unteraufgaben auf der Registerkarte Aufgaben überwachen und verwalten. Sie können den Fortschritt der Datenmigration verfolgen und Ihre Datenmigrationsaufgaben anhalten oder neu starten.

Ausgabe der Aufgabe zur Datenextraktion

Nach Abschluss der Migration sind Ihre Daten bereit. Lesen Sie die folgenden Informationen, um zu bestimmen, wie Sie abhängig vom gewählten Migrationsmodus und dem Speicherort Ihrer Daten fortfahren sollten.

Migrationsmodus	Speicherort der Daten
Extrahieren, hochladen und kopieren	Die Daten befinden sich bereits in Ihrem Amazon Redshift Data Warehouse. Sie können die Daten dort überprüfen und verwenden. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Clustern über Client-Tools und Code.
Extrahieren und hochladen	Die Extraktionsagenten haben Ihre Daten als Dateien in Ihrem Amazon S3 S3-Bucket gespeichert. Sie können den Amazon Redshift COPY-Befehl verwenden, um Ihre Daten in Amazon Redshift zu laden. Weitere Informationen finden Sie unter Daten aus Amazon S3 laden in der Amazon Redshift Redshift-Dokumentation. Ihr Amazon S3 S3-Bucket enthält mehrere Ordner, die den von Ihnen eingerichteten Extraktionsaufgaben entsprechen. Wenn Sie Ihre Daten in Amazon Redshift laden, geben Sie den Namen der von jeder Aufgabe erstellten Manifestdatei an. Die Manifestdatei wird im Aufgabenordner in Ihrem Amazon S3 S3-Bucket wie folgt angezeigt. 

Migrationsmodus	Speicherort der Daten
Nur extrahieren	Die Agenten für die Datenextraktion haben Ihre Daten als Dateien in Ihrem Arbeitsverzeichnis gespeichert. Kopieren Sie Ihre Daten manuell in Ihren Amazon S3 S3-Bucket und fahren Sie dann mit den Anweisungen für Extrahieren und Hochladen fort.

Verwenden Sie virtuelle Partitionierung mit AWS Schema Conversion Tool

Sie können große nicht partitionierte Tabellen häufig am besten verwalten, indem Sie Unteraufgaben anlegen, die virtuelle Partitionen der Tabellendaten mithilfe von Filterregeln erstellen. In AWS SCT können Sie virtuelle Partitionen für Ihre migrierten Daten erstellen. Es gibt drei Partitionstypen, die jeweils für bestimmte Datentypen eingesetzt werden:

- Der RANGE-Partitionstyp arbeitet mit numerischen Datentypen und Datum und Uhrzeit.
- Der LIST-Partitionstyp arbeitet mit numerischen Datentypen, Zeichen sowie Datum und Uhrzeit.
- Der Partitionstyp DATE AUTO SPLIT funktioniert mit numerischen, Datums- und Uhrzeitdatentypen.

AWS SCT validiert die Werte, die Sie für die Erstellung einer Partition angeben. Wenn Sie beispielsweise versuchen, eine Spalte mit dem Datentyp NUMERIC zu partitionieren, aber Werte eines anderen Datentyps angeben, wird ein AWS SCT Fehler ausgegeben.

Wenn Sie Daten AWS SCT zu Amazon Redshift migrieren, können Sie außerdem die native Partitionierung verwenden, um die Migration großer Tabellen zu verwalten. Weitere Informationen finden Sie unter Native Partitionierung verwenden.

Einschränkungen bei der Erstellung einer virtuellen Partitionierung

Dies sind Einschränkungen für die Erstellung einer virtuellen Partition:

- Sie können die virtuelle Partitionierung nur für partitionierte Tabellen verwenden.
- Sie können die virtuelle Partitionierung nur in der Datenmigration-Ansicht verwenden.
- Sie können die Option UNION ALL VIEW für die virtuelle Partitionierung nicht verwenden.

RANGE-Partitionstyp

Der Partitionstyp RANGE partitioniert Daten basierend auf eine Reihe von Spaltenwerten für numerische und Datums- und Uhrzeitdatentypen. Dieser Partitionstyp erstellt eine `WHERE`-Klausel, und geben Sie den Wertebereich für jede Partition an. Verwenden Sie das Feld Werte, um eine Werteliste für die partitionierte Spalte anzugeben. Sie können Werteinformationen über eine CSV-Datei laden.

Der RANGE-Partitionstyp erstellt Standardpartitionen an beiden Enden der Partitionswerte. Diese Standardpartitionen catch alle Daten ab, die kleiner oder größer als die angegebenen Partitionswerte sind.

Sie können beispielsweise mehrere Partitionen auf der Grundlage eines von Ihnen angegebenen Wertebereichs erstellen. Im folgenden Beispiel sind die Partitionierungswerte für LO_TAX so angegeben, dass mehrere Partitionen erstellt werden.

```
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
```

So erstellen Sie eine virtuelle RANGE-Partition

1. Öffnen AWS SCT.
2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die Informationen wie folgt ein.

Option	Aktion
Typ der Partition	Wählen Sie RANGE. Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Typ der Spalte	Wählen Sie den Datentyp für die Werte in der Spalte.

Option	Aktion
Werte	Fügen Sie neue Werte hinzu, indem Sie die einzelnen Werte in das Feld New Value eingeben, und klicken Sie auf das Pluszeichen, um die Werte hinzuzufügen.
Aus Datei laden	(Optional) Geben Sie den Namen einer .csv-Datei ein, die die Partitionswerte enthält.

5. Wählen Sie OK aus.

LIST-Partitionstyp

Der Partitionstyp LIST partitioniert Daten basierend auf Spaltenwerten für numerische, Zeichen- sowie Datums- und Uhrzeitdatentypen. Dieser Partitionstyp erstellt eine `WHERE`-Klausel, und Sie geben die Werte für jede Partition an. Verwenden Sie das Feld Werte, um eine Werteliste für die partitionierte Spalte anzugeben. Sie können Werteinformationen über eine CSV-Datei laden.

Sie können beispielsweise mehrere Partitionen basierend auf einem von Ihnen vorgegebenen Wert erstellen. Im folgenden Beispiel sind die Partitionierungswerte für LO_ORDERKEY so angegeben, dass mehrere Partitionen erstellt werden.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
…
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Sie können auch eine Standard-Partition für Werte erstellen, die nicht in den angegebenen enthalten sind.

Sie können den Partitionstyp LIST verwenden, um die Quelldaten zu filtern, wenn Sie bestimmte Werte von der Migration ausschließen möchten. Nehmen wir beispielsweise an, Sie möchten Zeilen mit `LO_ORDERKEY = 4` weglassen. Nehmen Sie in diesem Fall den Wert nicht `4` in die Liste der Partitionswerte auf und stellen Sie sicher, dass die Option Andere Werte einbeziehen nicht ausgewählt ist.

So erstellen Sie eine virtuelle LIST-Partition

1. Öffnen AWS SCT.

2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die Informationen wie folgt ein.

Option	Aktion
Typ der Partition	Klicken Sie auf LIST.... Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Neuer Wert	Geben Sie hier einen Wert ein, um ihn der Menge der Partition ierungswerte hinzuzufügen.
Andere Werte einbeziehen	Wählen Sie diese Option, um eine Standard-Partition zu erstellen, in der alle Werte gespeichert werden, die den Partitionierungskriterien nicht entsprechen.
Aus Datei laden	(Optional) Geben Sie den Namen einer .csv-Datei ein, die die Partitionswerte enthält.

5. Wählen Sie OK aus.

DATE AUTO SPLIT Partitionstyp

Der Partitionstyp DATE AUTO SPLIT ist eine automatisierte Methode zum Generieren von RANGE-Partitionen. Mit DATA AUTO SPLIT teilen Sie AWS SCT dem Partitionierungsattribut mit, wo es beginnen und enden soll und wie groß der Bereich zwischen den Werten ist. AWS SCT Berechnet dann automatisch die Partitionswerte.

DATA AUTO SPLIT automatisiert einen Großteil der Arbeit, die mit der Erstellung von Bereichspartitionen verbunden ist. Der Kompromiss zwischen der Verwendung dieser Technik und der Bereichspartitionierung besteht darin, wie viel Kontrolle Sie über die Partitionsgrenzen benötigen. Beim automatischen Aufteilen werden immer Bereiche gleicher Größe (einheitlich) erstellt. Durch die Bereichspartitionierung können Sie die Größe der einzelnen Bereiche je nach Bedarf für Ihre spezielle Datenverteilung variieren. Sie können sie beispielsweise täglich, wöchentlich, zweiwöchentlich, monatlich usw. verwenden.

```
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’
```

So erstellen Sie eine virtuelle DATE AUTO SPLIT-Partition

1. Öffnen AWS SCT.
2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die folgenden Informationen ein.

Option	Aktion
Typ der Partition	Wählen Sie DATE AUTO SPLIT. Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Startdatum	Geben Sie ein Anfangsdatum ein.
Enddatum	Geben Sie ein Enddatum ein.
Interval	Geben Sie die Intervalleinheit ein und wählen Sie den Wert für diese Einheit.

5. Wählen Sie OK aus.

Native Partitionierung verwenden

Um die Datenmigration zu beschleunigen, können Ihre Datenextraktionsagenten native Partitionen von Tabellen auf Ihrem Data Warehouse-Quellserver verwenden. AWS SCT unterstützt native Partitionierung für Migrationen von Greenplum, Netezza und Oracle zu Amazon Redshift.

Nachdem Sie ein Projekt erstellt haben, können Sie beispielsweise Statistiken zu einem Schema sammeln und die Größe der für die Migration ausgewählten Tabellen analysieren. Bei Tabellen, die

die angegebene Größe überschreiten, wird der systemeigene Partitionierungsmechanismus AWS SCT ausgelöst.

Um die native Partitionierung zu verwenden

1. Öffnen AWS SCT Sie und wählen Sie Neues Projekt für Datei. Das Dialogfeld „Neues Projekt“ wird angezeigt.
2. Erstellen Sie ein neues Projekt, fügen Sie Ihre Quell- und Zielserver hinzu und erstellen Sie Zuordnungsregeln. Weitere Informationen finden Sie unter Projekte starten und verwalten in AWS SCT.
3. Wählen Sie Ansicht und anschließend Hauptansicht.
4. Wählen Sie für die Projekteinstellungen die Registerkarte Datenmigration. Wählen Sie Automatische Partitionierung verwenden. Geben Sie für Greenplum- und Netezza-Quelldatenbanken die Mindestgröße der unterstützten Tabellen in Megabyte ein (z. B. 100). AWS SCT erstellt automatisch separate Migrationsunteraufgaben für jede native Partition, die nicht leer ist. AWS SCT Erstellt für Migrationen von Oracle zu Amazon Redshift Unteraufgaben für alle partitionierten Tabellen.
5. Wählen Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, ein Schema aus. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie Statistik sammeln. Bei der Datenmigration von Oracle zu Amazon Redshift können Sie diesen Schritt überspringen.
6. Wählen Sie alle zu migrierenden Tabellen aus.
7. Registrieren Sie die erforderliche Anzahl von Agenten. Weitere Informationen finden Sie unter Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool.
8. Erstellen Sie eine Datenextraktionsaufgabe für die ausgewählten Tabellen. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen.

   Prüfen Sie, ob große Tabellen in Unteraufgaben aufgeteilt sind und ob jede Unteraufgabe dem Datensatz entspricht, der einen Teil der Tabelle darstellt, der sich in einem Bereich in Ihrem Quell-Data Warehouse befindet.
9. Starten und überwachen Sie den Migrationsprozess, bis die AWS SCT Datenextraktionsagenten die Migration der Daten aus Ihren Quelltabellen abgeschlossen haben.

Migration LOBs zu Amazon Redshift

Amazon Redshift unterstützt das Speichern großer binärer Objekte (LOBs) nicht. Wenn Sie jedoch eine oder mehrere LOBs zu Amazon Redshift migrieren müssen, AWS SCT können Sie die Migration durchführen. AWS SCT Verwendet dazu einen Amazon S3 S3-Bucket zum Speichern von LOBs und schreibt die URL für den Amazon S3 S3-Bucket in die migrierten Daten, die in Amazon Redshift gespeichert sind.

Um zu Amazon LOBs Redshift zu migrieren

1. Öffnen Sie ein AWS SCT Projekt.
2. Stellen Sie eine Verbindung mit der Quell- und Zieldatenbank her. Aktualisieren Sie die Metadaten aus der Zieldatenbank und stellen Sie sicher, dass die konvertierten Tabellen dort vorhanden sind.
3. Wählen Sie unter Aktionen die Option Lokale Aufgabe erstellen aus.
4. Wählen Sie für den Migrationsmodus eine der folgenden Optionen:

   - Extrahieren und Hochladen, um Ihre Daten zu extrahieren, und laden Sie Ihre Daten auf Amazon S3 hoch.
   - Extrahieren, Hochladen und Kopieren, um Ihre Daten zu extrahieren, laden Sie Ihre Daten auf Amazon S3 hoch und kopieren Sie sie in Ihr Amazon Redshift Data Warehouse.
5. Wählen Sie Amazon S3 S3-Einstellungen.
6. Geben Sie für den Amazon S3 LOBs S3-Bucket-Ordner den Namen des Ordners in einem Amazon S3 S3-Bucket ein, in dem Sie den LOBs speichern möchten.

   Wenn Sie ein AWS Serviceprofil verwenden, ist dieses Feld optional. AWS SCT kann die Standardeinstellungen Ihres Profils verwenden. Um einen anderen Amazon S3 S3-Bucket zu verwenden, geben Sie den Pfad hier ein.
7. Aktivieren Sie die Option Proxy verwenden, um einen Proxy-Server zum Hochladen von Daten auf Amazon S3 zu verwenden. Wählen Sie dann das Datenübertragungsprotokoll und geben Sie den Hostnamen, den Port, den Benutzernamen und das Passwort ein.
8. Wählen Sie als Endpunkttyp FIPS aus, um den FIPS-Endpunkt (Federal Information Processing Standard) zu verwenden. Wählen Sie VPCE, um den Virtual Private Cloud (VPC) -Endpunkt zu verwenden. Geben Sie dann für VPC-Endpunkt das Domain Name System (DNS) Ihres VPC-Endpunkts ein.

9. Aktivieren Sie die Option Dateien nach dem Kopieren nach Amazon Redshift auf Amazon S3 behalten, um die extrahierten Dateien nach dem Kopieren dieser Dateien nach Amazon Redshift auf Amazon S3 beizubehalten.
10. Klicken Sie auf Create, um die Aufgabe zu erstellen.

Bewährte Methoden und Problembehebung für Datenextraktionsagenten

Nachfolgend finden Sie einige bewährte Methoden und Problemlösungen für die Verwendung von Agenten für die Datenextraktion.

Problem	Vorschläge für die Fehlerbehebung
Die Leistung ist langsam.	Wir empfehlen zur Leistungsverbesserung Folgendes: • Installieren Sie mehrere Agenten. • Installieren Sie Agenten auf Computern, die sich in der Nähe Ihres Data Warehouses befinden. • Bearbeiten Sie nicht alle Tabellen in einer einzelnen Agentenaufgabe.
Verzögerungen durch Konflikte	Vermeiden Sie es, dass zu viele Agenten gleichzeitig auf Ihr Data Warehouse zugreifen.
Ein Agent fällt vorübergehend aus.	Wenn ein Agent ausfällt, wird der Status seiner Aufgaben in AWS SCT als fehlgeschlagen angezeigt. Wenn Sie warten, kann sich der Agent möglicherweise fangen. In diesem Fall wird der Status seiner Aufgaben in AWS SCT aktualisiert.
Ein Agent fällt dauerhaft aus.	Wenn der Computer, auf dem ein Agent ausgeführt wird, dauerhaft ausfällt und der Agent gerade eine Aufgabe ausführt, können Sie den Agenten austauschen und den neuen Agenten mit der Aufgabe betrauen. Dies ist nur möglich, wenn das Arbeitsverzeichnis des ursprünglichen Agenten sich nicht auf demselben Computer wie der ursprüngliche Agent befindet. Gehen Sie wie folgt vor, um einen Agenten auszutauschen:

Problem	Vorschläge für die Fehlerbehebung
	• Installieren Sie einen Agenten auf einem neuen Computer. • Konfigurieren Sie den neuen Agenten mit den gleichen Einstellungen wie Portnummer und Arbeitsverzeichnis wie den ursprünglichen Agenten. • Starten Sie den Agenten. Nachdem Start erkennt die Aufgabe den neuen verfügbaren Agenten und fährt mit dem neuen Agenten fort.

Anwendungs-SQL konvertieren mit AWS SCT

Wenn Sie Ihr Datenbankschema von einer Engine in eine andere konvertieren, müssen Sie auch den SQL-Code in Ihren Anwendungen aktualisieren, damit diese mit der neuen Datenbank-Engine anstelle der alten interagieren. Sie können den konvertierten SQL-Code anzeigen, analysieren, bearbeiten und speichern.

Sie können die AWS Schema Conversion Tool (AWS SCT) verwenden, um den SQL-Code in Ihren C++-, C#-, Java- oder anderen Anwendungscode zu konvertieren. Für eine Konvertierung von Oracle nach PostgreSQL können Sie verwenden, um SQL*Plus-Code in PSQL AWS SCT zu konvertieren. Bei einer Konvertierung von Oracle nach PostgreSQL können Sie auch SQL-Code AWS SCT konvertieren, der in C#-, C++-, Java- und Pro*C-Anwendungen eingebettet ist.

Themen

- Überblick über die Konvertierung von Anwendungs-SQL
- Konvertieren von SQL-Code in Ihren Anwendungen mit AWS SCT
- Konvertieren von SQL-Code in C#-Anwendungen mit AWS Schema Conversion Tool
- Konvertieren von SQL-Code in C++-Anwendungen mit AWS Schema Conversion Tool
- Konvertieren von SQL-Code in Java-Anwendungen mit AWS Schema Conversion Tool
- Konvertierung von SQL-Code in Pro*C-Anwendungen mit AWS Schema Conversion Tool

Überblick über die Konvertierung von Anwendungs-SQL

Führen Sie die folgenden allgemeinen Schritte aus, um den SQL-Code in Ihrer Anwendung zu konvertieren:

- Erstellen Sie ein Anwendungskonvertierungsprojekt — Das Anwendungskonvertierungsprojekt ist ein untergeordnetes Element des Projekts zur Konvertierung des Datenbankschemas. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendungskonvertierungsprojekt enthalten. Weitere Informationen finden Sie unter Generische Projekte zur Konvertierung von Anwendungen erstellen in AWS SCT.
- Analysieren und konvertieren Sie Ihren SQL-Code — AWS SCT analysiert Ihre Anwendung, extrahiert den SQL-Code und erstellt eine lokale Version der konvertierten SQL, die Sie überprüfen und bearbeiten können. Das Tool ändert den Code in Ihrer Anwendung erst dann, wenn Sie bereit

sind. Weitere Informationen finden Sie unter Analysieren und konvertieren Sie Ihren SQL-Code in AWS SCT.

- Erstellen Sie einen Anwendungsbewertungsbericht — Der Anwendungsbewertungsbericht enthält wichtige Informationen zur Konvertierung des Anwendungs-SQL-Codes von Ihrem Quelldatenbankschema in Ihr Zieldatenbankschema. Weitere Informationen finden Sie unter Erstellen und Verwenden des AWS SCT Bewertungsberichts in AWS SCT.
- Bearbeiten Sie Ihren konvertierten SQL-Code, wenden Sie Änderungen an und speichern Sie ihn. Der Bewertungsbericht enthält eine Liste von SQL-Codeelementen, die nicht automatisch konvertiert werden können. Für diese Elemente können Sie den SQL-Code manuell bearbeiten, um die Konvertierung durchzuführen. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihres konvertierten SQL-Codes mit AWS SCT.

Konvertieren von SQL-Code in Ihren Anwendungen mit AWS SCT

Sie können AWS SCT es verwenden, um in Ihre Anwendungen eingebetteten SQL-Code zu konvertieren. Der generische AWS SCT Anwendungskonverter behandelt Ihren Anwendungscode als einfachen Text. Er scannt Ihren Anwendungscode und extrahiert SQL-Code mit regulären Ausdrücken. Dieser Konverter unterstützt verschiedene Arten von Quellcodedateien und arbeitet mit Anwendungscode, der in einer beliebigen Programmiersprache geschrieben ist.

Der generische Anwendungskonverter hat die folgenden Einschränkungen. Er befasst sich nicht eingehend mit der Anwendungslogik, die für die Programmiersprache Ihrer Anwendung spezifisch ist. Außerdem unterstützt der generische Konverter keine SQL-Anweisungen von verschiedenen Anwendungsobjekten wie Funktionen, Parametern, lokalen Variablen usw.

Verwenden Sie sprachspezifische Anwendungs-SQL-Code-Konverter, um die Konvertierung von Anwendungs-SQL-Code zu verbessern. Weitere Informationen finden Sie unter SQL-Code in C#-Anwendungen, SQL-Code in Java und SQL-Code in Pro*C.

Generische Projekte zur Konvertierung von Anwendungen erstellen in AWS SCT

In der AWS Schema Conversion Tool ist das Anwendungskonvertierungsprojekt ein untergeordnetes Element des Datenbankschema-Konvertierungsprojekts. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendungskonvertierungsprojekt enthalten.

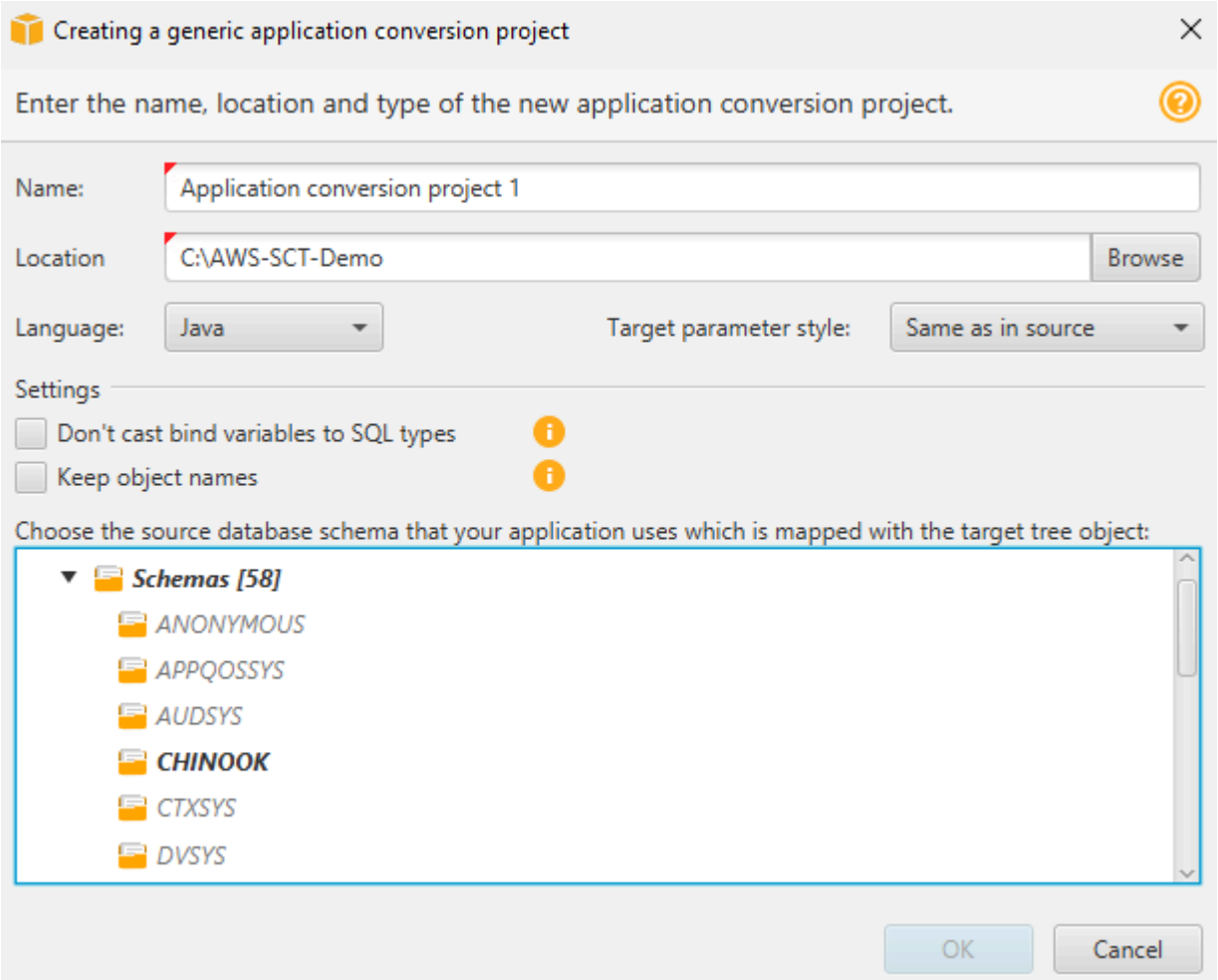
> **Note**
>
> AWS SCT unterstützt keine Konvertierung zwischen den folgenden Quellen und Zielen:
>
> - Von Oracle nach Oracle
> - PostgreSQL nach PostgreSQL oder Aurora PostgreSQL
> - MySQL nach MySQL
> - SQL Server zu SQL Server
> - Amazon Redshift zu Amazon Redshift
> - SQL Server zu Babelfish
> - SQL Server-Integrationsdienste für AWS Glue
> - Apache Cassandra nach Amazon DynamoDB

Gehen Sie wie folgt vor, um ein Projekt zur Konvertierung generischer Anwendungen zu erstellen.

So erstellen Sie ein Anwendungskonvertierungsprojekt

1. Wählen Sie in der AWS Schema Conversion Tool im Menü Anwendungen die Option Neue generische Anwendung aus.

   Das Dialogfeld New application conversion project wird angezeigt.



2. Fügen Sie die folgenden Projektinformationen hinzu.

Für diesen Parameter	Vorgehensweise
Name	Geben Sie einen Namen für Ihr Projekt zur Anwendung skonvertierung ein. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendung skonvertierungsprojekt aufweisen. Wählen Sie also einen Namen, der sinnvoll ist, wenn Sie zu einem späteren Zeitpunkt weitere Projekte hinzufügen.
Ort	Geben Sie den Speicherort des Quellcodes für Ihre Anwendung ein.
Sprache	Wählen Sie eine der folgenden Optionen aus:

Für diesen Parameter	Vorgehensweise
	• Java • C++ • C# • Alle
Stil des Zielparameters	Wählen Sie die Syntax, die für Bind-Variablen im konvertierten Code verwendet werden soll. Verschiedene Datenbankplattformen verwenden unterschiedliche Syntax für Bind-Variablen. Wählen Sie eine der folgenden Optionen: • Same as in source • Positional (?) • Indexed (:1) • Indexed ($1) • Named (@name) • Named (:name) • Benannt (&Name) • Benannt ($name) • Benannt (#name) • Benannt (! Name!)
Wählen Sie das Quelldatenbankschema	Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist.

3. Wählen Sie Bindungsvariablen nicht in SQL-Typen umwandeln aus, um eine Konvertierung von Bindungsvariablentypen in SQL-Typen zu vermeiden. Diese Option ist nur für eine Konvertierung von Oracle nach PostgreSQL verfügbar.

   Ihr Quellanwendungscode enthält beispielsweise die folgende Oracle-Abfrage:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Wenn Sie „Bind-Variablen nicht in SQL-Typen umwandeln“ auswählen, wird diese Abfrage wie folgt AWS SCT konvertiert.

```
SELECT * FROM account WHERE id = ?
```

Wenn Sie die Option Bindungsvariablen nicht in SQL-Typen umwandeln deaktivieren, wird der Bindungsvariablentyp in den `NUMERIC` Datentyp AWS SCT geändert. Das Konvertierungsergebnis wird im Folgenden dargestellt.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

4. Wählen Sie Objektnamen beibehalten, um zu vermeiden, dass der Schemaname zum Namen des konvertierten Objekts hinzugefügt wird. Diese Option ist nur für eine Konvertierung von Oracle nach PostgreSQL verfügbar.

   Nehmen wir beispielsweise an, dass Ihr Quellanwendungscode die folgende Oracle-Abfrage enthält.

   ```
   SELECT * FROM ACCOUNT
   ```

   Wenn Sie Objektnamen beibehalten auswählen, wird diese Abfrage wie folgt AWS SCT konvertiert.

   ```
   SELECT * FROM account
   ```

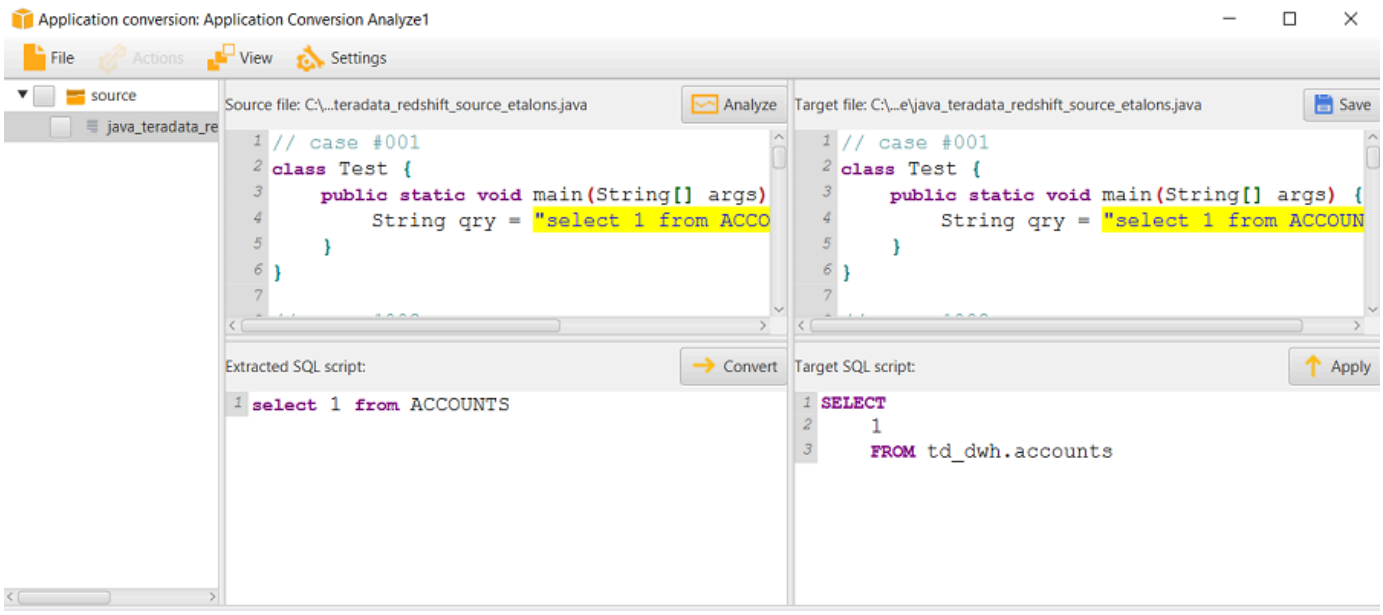
   Wenn Sie die Option Objektnamen beibehalten deaktivieren, wird der Schemaname zum Namen der Tabelle AWS SCT hinzugefügt. Das Konvertierungsergebnis wird im Folgenden dargestellt.

   ```
   SELECT * FROM schema_name.account
   ```

   Wenn Ihr Quellcode die Namen der übergeordneten Objekte in den Namen der Objekte enthält, AWS SCT verwendet er dieses Format im konvertierten Code. Ignorieren Sie in diesem Fall die Option Objektnamen beibehalten, da die Namen der übergeordneten Objekte dem konvertierten Code AWS SCT hinzugefügt werden.

5. Klicken Sie auf OK, um Ihr Anwendungskonvertierungsprojekt zu erstellen.

   Das Projektfenster wird geöffnet.



Verwaltung von Projekten zur Anwendungskonvertierung in AWS SCT

Sie können ein vorhandenes Projekt zur Anwendungskonvertierung öffnen und mehrere Projekte zur Anwendungskonvertierung hinzufügen.

Nachdem Sie ein Projekt zur Anwendungskonvertierung erstellt haben, wird das Projektfenster automatisch geöffnet. Sie können das Projektfenster für die Anwendungskonvertierung schließen und später wieder darauf zugreifen.

Um ein vorhandenes Projekt zur Anwendungskonvertierung zu öffnen

1. Wählen Sie im linken Bereich den Projektknoten für die Anwendungskonvertierung aus und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie „Anwendung verwalten“.

Um ein zusätzliches Projekt zur Anwendungskonvertierung hinzuzufügen

1. Wählen Sie im linken Bereich den Projektknoten für die Anwendungskonvertierung aus und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie New application (Neue Anwendung) aus.
3. Geben Sie die Informationen ein, die für die Erstellung eines neuen Projekts zur Anwendungskonvertierung erforderlich sind. Weitere Informationen finden Sie unter Projekte zur Konvertierung generischer Anwendungen erstellen.

Analysieren und konvertieren Sie Ihren SQL-Code in AWS SCT

Verwenden Sie das folgende Verfahren, um Ihren SQL-Code in der zu analysieren und zu konvertieren AWS Schema Conversion Tool.

So analysieren und konvertieren Sie den SQL-Code

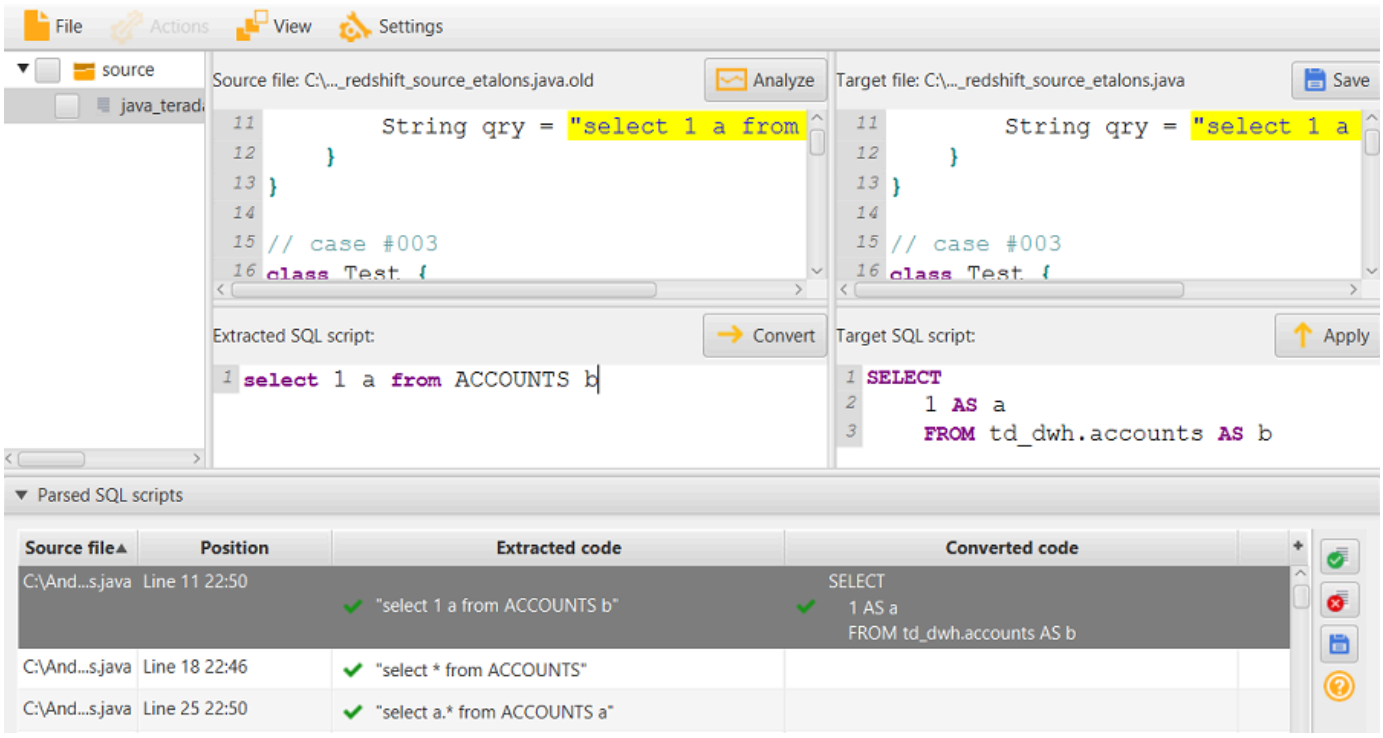
1. Öffnen Sie ein vorhandenes Projekt zur Anwendungskonvertierung und wählen Sie Analysieren aus.

   AWS SCT analysiert Ihren Anwendungscode und extrahiert den SQL-Code. AWS SCT zeigt den extrahierten SQL-Code in der Liste der analysierten SQL-Skripten an.

2. Wählen Sie für geparste SQL-Skripts ein Element aus, um den extrahierten SQL-Code zu überprüfen. AWS SCT zeigt den Code des ausgewählten Elements im Bereich Extrahiertes SQL-Skript an.

3. Wählen Sie „Konvertieren“, um den SQL-Code im Bereich „Extrahiertes SQL-Skript“ zu konvertieren. AWS SCT konvertiert den Code in ein Format, das mit Ihrer Zieldatenbank kompatibel ist.

   Sie können den konvertierten SQL-Code bearbeiten. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihres konvertierten SQL-Codes.



4. Wenn Sie einen Bewertungsbericht zur Anwendungskonvertierung erstellen, werden alle extrahierten SQL-Codeelemente AWS SCT konvertiert. Weitere Informationen finden Sie unter Erstellung und Verwendung des Bewertungsberichts.

Erstellen und Verwenden des AWS SCT Bewertungsberichts in AWS SCT

Der Bewertungsbericht zur Anwendungskonvertierung enthält Informationen zur Konvertierung des Anwendungs-SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bericht beschreibt den gesamten extrahierten SQL-Code, den gesamten konvertierten SQL-Code und die Aktionspunkte für SQL-Code, der nicht konvertiert AWS SCT werden kann.

Erstellen eines Bewertungsberichts zur Anwendungskonvertierung

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Anwendungskonvertierung zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Anwendungskonvertierung

1. Wählen Sie im Projektfenster zur Anwendungskonvertierung im Menü Aktionen die Option Bericht erstellen aus.

   AWS SCT erstellt den Bewertungsbericht zur Anwendungskonvertierung und öffnet ihn im Projektfenster für die Anwendungskonvertierung.
2. Prüfen Sie die Registerkarte Summary.

   Auf der Registerkarte Zusammenfassung (siehe unten) werden die zusammenfassenden Informationen aus dem Bewertungsbericht für die Anwendung angezeigt. Es werden sowohl die automatisch konvertierten SQL-Codeelemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.



3. Wählen Sie SQL-Extraktionsaktionen aus.

   Sehen Sie sich die Liste der SQL-Codeelemente an, die nicht aus Ihrem Quellcode extrahiert werden AWS SCT können.
4. Wählen Sie SQL-Konvertierungsaktionen aus.

Sehen Sie sich die Liste der SQL-Codeelemente an, die nicht automatisch konvertiert werden AWS SCT können. Verwenden Sie die empfohlenen Aktionen, um den SQL-Code manuell zu konvertieren. Hinweise zur Bearbeitung des konvertierten SQL-Codes finden Sie unterBearbeiten und Speichern Ihres konvertierten SQL-Codes mit AWS SCT.

5. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Wählen Sie oben rechts die Option Als PDF speichern, um den Bericht als PDF-Datei zu speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.
   - Wählen Sie oben rechts die Option In CSV speichern, um den Bericht als CSV-Datei zu speichern.

     Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Bearbeiten und Speichern Ihres konvertierten SQL-Codes mit AWS SCT

Der Bewertungsbericht enthält eine Liste von SQL-Codeelementen, die nicht konvertiert AWS SCT werden können. AWS SCT Erstellt für jedes Element einen Aktionspunkt auf der Registerkarte „SQL-Konvertierungsaktionen“. Für diese Elemente können Sie den SQL-Code manuell bearbeiten, um die Konvertierung durchzuführen.

Führen Sie die folgenden Schritte aus, um den konvertierten SQL-Code zu bearbeiten, die Änderungen zu übernehmen und sie anschließend zu speichern.

So können Sie konvertierten SQL-Code bearbeiten, Änderungen daran übernehmen und ihn speichern

1. Bearbeiten Sie den konvertierten SQL-Code direkt im Bereich Target SQL script. Wenn kein konvertierter Code angezeigt wird, können Sie auf den Bereich klicken und mit der Eingabe beginnen.

2. Nachdem Sie die Bearbeitung des konvertierten SQL-Codes abgeschlossen haben, wählen Sie Apply aus. An diesem Punkt werden die Änderungen im Arbeitsspeicher gespeichert, aber noch nicht in Ihre Datei geschrieben.
3. Wählen Sie Save aus, um Ihre Änderungen in Ihrer Datei zu speichern.

   Wenn Sie Speichern wählen, überschreiben Sie Ihre Originaldatei. Erstellen Sie vor dem Speichern eine Kopie der ursprünglichen Datei, sodass Sie einen Datensatz des ursprünglichen Anwendungscodes haben.

Konvertieren von SQL-Code in C#-Anwendungen mit AWS Schema Conversion Tool

Für eine Konvertierung von Oracle nach PostgreSQL können Sie AWS Schema Conversion Tool (AWS SCT) verwenden, um in Ihre C#-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle C#-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, z. B. Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für C#-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Erstellen von Projekten zur Konvertierung von C#-Anwendungen in AWS SCT

Sie können ein C#-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.

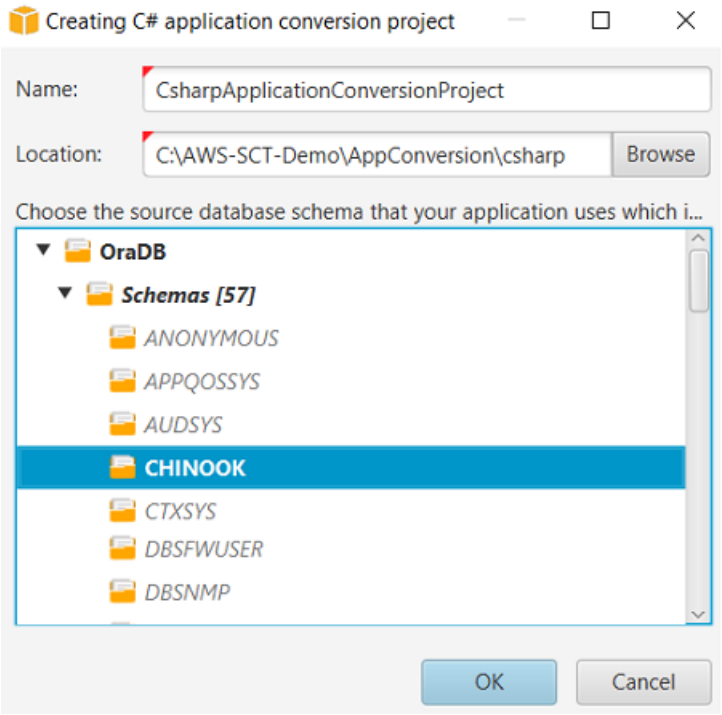
Sie können mehrere Projekte zur Anwendungskonvertierung in einem einzigen AWS SCT Projekt hinzufügen. Gehen Sie wie folgt vor, um ein C#-Anwendungskonvertierungsprojekt zu erstellen.

So erstellen Sie ein Projekt zur Konvertierung von C#-Anwendungen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter Projekte starten und verwalten in AWS SCT und Server zum Projekt hinzufügen in AWS SCT.

2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool und Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
3. Wählen Sie im Menü Ansicht die Option Hauptansicht.
4. Wählen Sie im Menü Anwendungen die Option Neue C#-Anwendung aus.

   Das Dialogfeld „Projekt zur Konvertierung einer C#-Anwendung erstellen“ wird angezeigt.



5. Geben Sie unter Name einen Namen für Ihr C#-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Geben Sie unter Standort den Speicherort des Quellcodes für Ihre Anwendung ein.
7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
8. Wählen Sie OK, um Ihr C#-Anwendungskonvertierungsprojekt zu erstellen.
9. Suchen Sie Ihr Projekt zur Konvertierung von C#-Anwendungen im linken Bereich im Knoten Anwendungen.

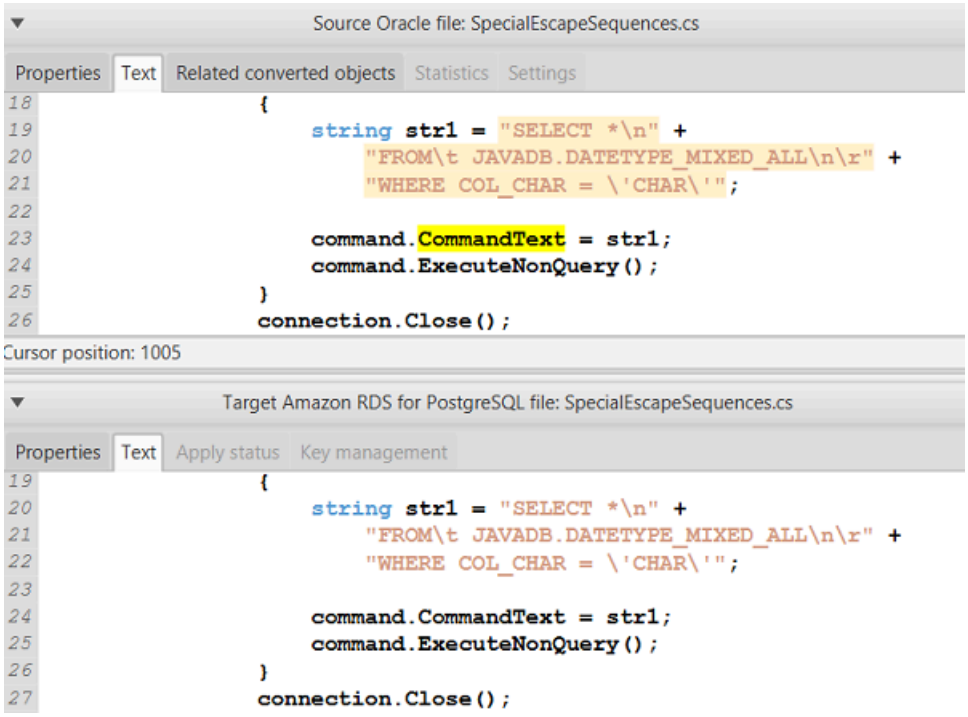
Konvertieren Sie den SQL-Code Ihrer C#-Anwendung in AWS SCT

Nachdem Sie Ihre C#-Anwendung zum AWS SCT Projekt hinzugefügt haben, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre C#-Anwendung eingebetteten SQL-Code in der zu analysieren und zu konvertieren. AWS Schema Conversion Tool

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie den C#-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“. AWS SCT analysiert Ihre Quellcodedateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen C#-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

   AWS SCT Erstellt im Bereich Zieldatenbank eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.



4. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter Speichern Sie Ihren konvertierten Anwendungscode.

Ihre C#-Anwendungen können SQL-Code enthalten, der mit verschiedenen Quelldatenbanken interagiert. Sie können mehrere dieser Quelldatenbanken zu PostgreSQL migrieren. Stellen Sie in diesem Fall sicher, dass Sie keinen SQL-Code konvertieren, der mit Datenbanken interagiert, die Sie aus dem Migrationsbereich ausgeschlossen haben. Sie können Quelldateien Ihrer C#-Anwendung vom Konvertierungsbereich ausschließen. Deaktivieren Sie dazu die Kontrollkästchen für die Namen der Dateien, die Sie aus dem Konvertierungsbereich ausschließen möchten.

Nachdem Sie den Konvertierungsbereich geändert haben, analysiert es AWS SCT weiterhin den SQL-Code aller Quelldateien Ihrer C#-Anwendungen. AWS SCT Kopiert dann alle Quelldateien, die Sie aus dem Konvertierungsbereich ausgeschlossen haben, in den Zielordner. Dieser Vorgang ermöglicht es, Ihre Anwendung zu erstellen, nachdem Sie Ihre konvertierten Anwendungsdateien gespeichert haben.

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Gehen Sie wie folgt vor, um den konvertierten Anwendungscode zu speichern.

Um den konvertierten Anwendungscode zu speichern

1. Erweitern Sie den C#-Knoten unter Anwendungen im Bereich Zieldatenbank.
2. Wählen Sie Ihre konvertierte Anwendung aus und klicken Sie auf Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Ordner auswählen.

Verwaltung von Projekten zur Konvertierung von C#-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von C#-Anwendungen hinzufügen, den Anwendungscode im AWS SCT Projekt aktualisieren oder ein C#-Konvertierungsprojekt aus Ihrem Projekt entfernen. AWS SCT

Um ein zusätzliches Projekt zur Konvertierung von C#-Anwendungen hinzuzufügen

1. Erweitern Sie den Knoten Anwendungen im linken Bereich.
2. Wählen Sie den C#-Knoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.

4. Geben Sie die Informationen ein, die zum Erstellen eines neuen Projekts zur Konvertierung von C#-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter Projekte zur Konvertierung von C#-Anwendungen erstellen.

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in das AWS SCT Projekt hoch.

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie den C#-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu aktualisierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Aktualisieren“ und dann „Ja“.

   AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um die Codeänderungen, die Sie in vorgenommen haben, AWS SCT und die Konvertierungsergebnisse beizubehalten, erstellen Sie ein neues C#-Konvertierungsprojekt.

Um ein C#-Anwendungskonvertierungsprojekt zu entfernen

1. Erweitern Sie den C#-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu entfernende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Löschen und dann OK.

Erstellen Sie einen Bewertungsbericht zur C#-Anwendungskonvertierung in AWS SCT

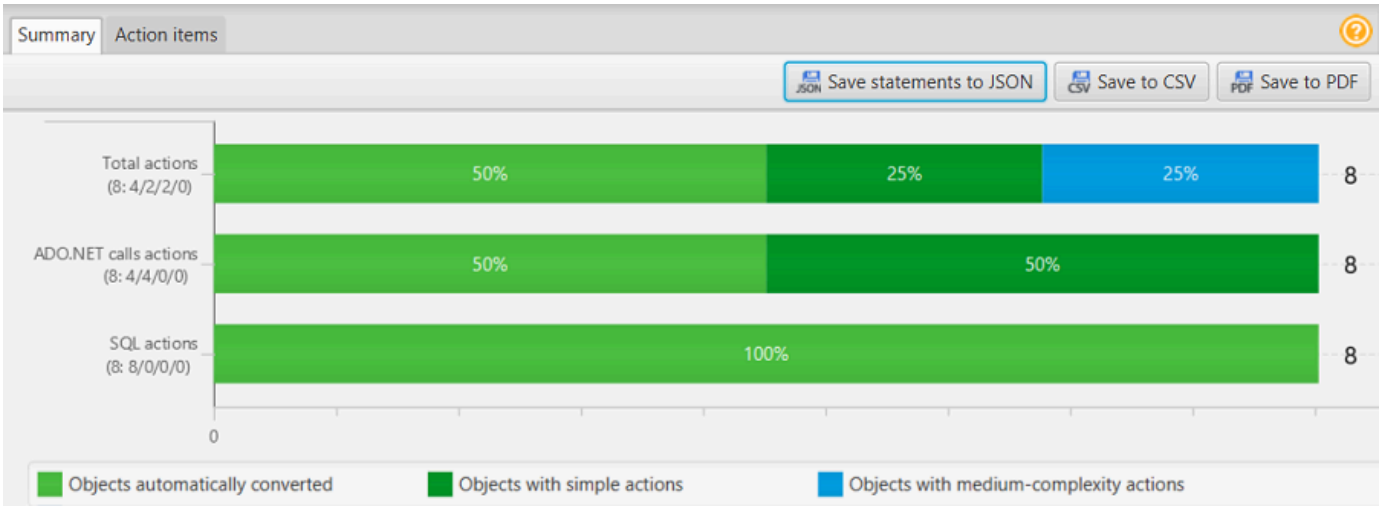
Der Bewertungsbericht zur C#-Anwendungskonvertierung enthält Informationen zur Konvertierung des in Ihre C#-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, der nicht konvertiert AWS SCT werden kann.

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur C#-Anwendungskonvertierung zu erstellen.

So erstellen Sie einen Bewertungsbericht zur C#-Anwendungskonvertierung

1. Erweitern Sie im linken Bereich den C#-Knoten unter Anwendungen.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“.
4. Wählen Sie im Menü „Ansicht“ die Option „Bewertungsbericht anzeigen“.
5. Rufen Sie die Registerkarte „Zusammenfassung“ auf.

   Auf der Registerkarte Zusammenfassung (siehe unten) werden die zusammenfassenden Informationen aus dem Bericht zur Bewertung der C#-Anwendung angezeigt. Es werden Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien angezeigt.



6. Wählen Sie Anweisungen in JSON speichern, um den extrahierten SQL-Code aus Ihrer C#-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Wählen Sie oben rechts die Option Als PDF speichern, um den Bericht als PDF-Datei zu speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.
   - Wählen Sie oben rechts die Option In CSV speichern, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertieren von SQL-Code in C++-Anwendungen mit AWS Schema Conversion Tool

Für eine Konvertierung von Oracle nach PostgreSQL können AWS SCT Sie den in Ihre C++-Anwendungen eingebetteten SQL-Code konvertieren. Dieser spezielle C++-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, wie Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für C++-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Projekte zur Konvertierung von C++-Anwendungen erstellen in AWS SCT

Sie können ein C++-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.

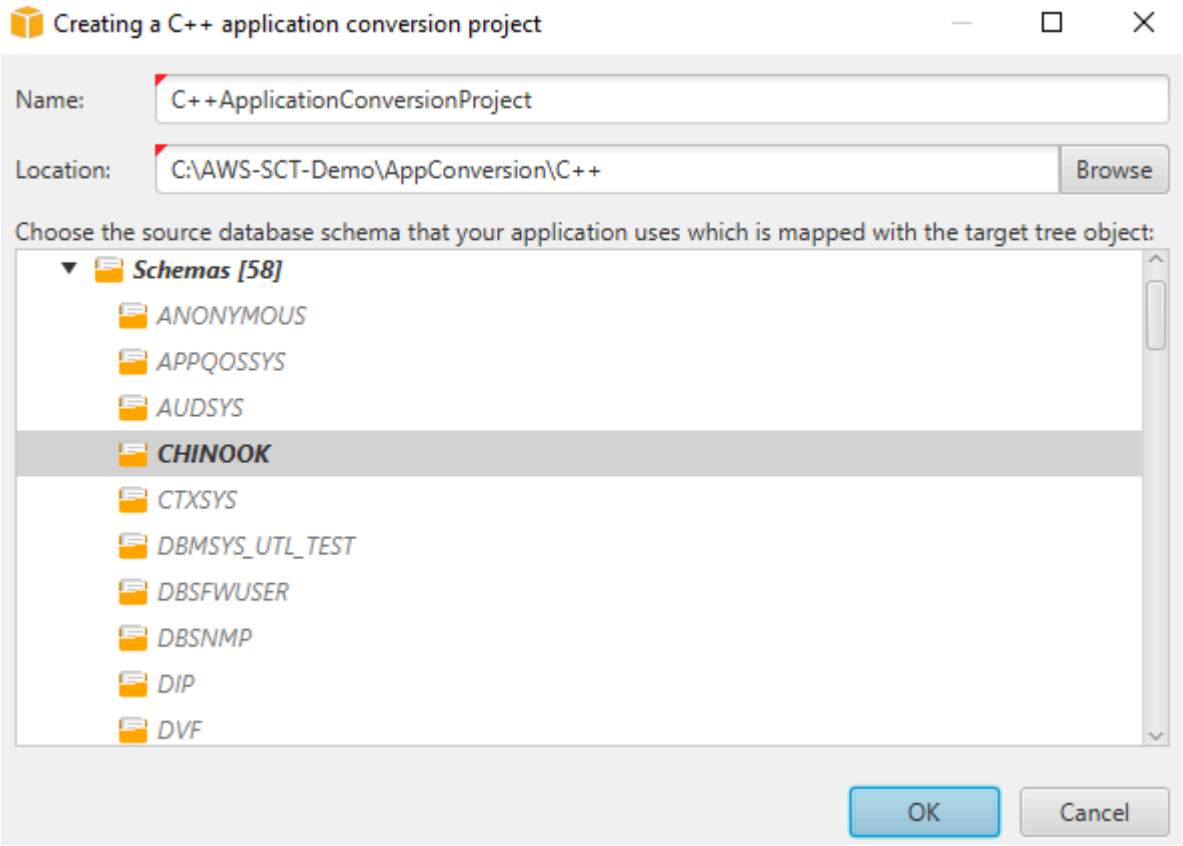
Sie können mehrere Projekte zur Anwendungskonvertierung in einem einzigen AWS SCT Projekt hinzufügen.

Um ein Projekt zur Konvertierung von C++-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter Projekte starten und verwalten in AWS SCT und Server zum Projekt hinzufügen in AWS SCT.
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool und Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.

3. Wählen Sie im Menü Ansicht die Option Hauptansicht.
4. Wählen Sie im Menü Anwendungen die Option Neue C++-Anwendung aus.

   Das Dialogfeld „Projekt zur Konvertierung einer C++-Anwendung erstellen“ wird angezeigt.



5. Geben Sie unter Name einen Namen für Ihr C++-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Geben Sie unter Standort den Speicherort des Quellcodes für Ihre Anwendung ein.
7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
8. Wählen Sie OK, um Ihr C++-Anwendungskonvertierungsprojekt zu erstellen.
9. Suchen Sie Ihr Projekt zur Konvertierung von C++-Anwendungen im linken Bereich im Knoten Anwendungen.

Konvertieren Sie den SQL-Code Ihrer C++-Anwendung in AWS SCT

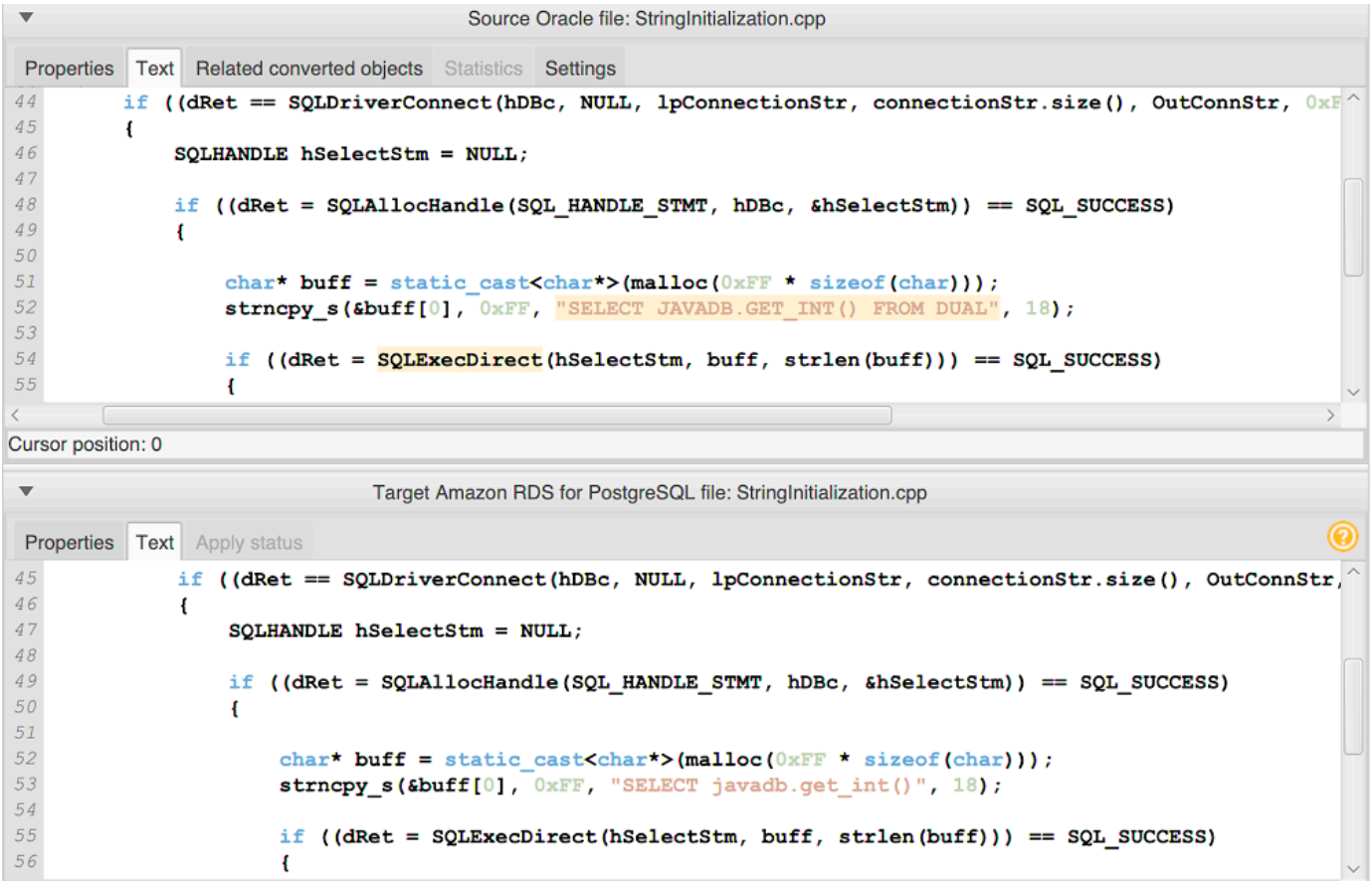
Nachdem Sie Ihre C++-Anwendung zum AWS SCT Projekt hinzugefügt haben, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihrer C++-Anwendung eingebetteten SQL-Code zu analysieren und zu konvertieren AWS SCT.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie im linken Bereich den C++-Knoten unter Anwendungen und wählen Sie die zu konvertierende Anwendung aus.
2. Wählen Sie im Projekt Source Oracle Application die Option Einstellungen aus. Überprüfen und bearbeiten Sie die Konvertierungseinstellungen für die ausgewählte C++-Anwendung. Sie können auch die Konvertierungseinstellungen für alle C++-Anwendungen angeben, die Sie Ihrem AWS SCT Projekt hinzugefügt haben. Weitere Informationen finden Sie unter Verwaltung von Projekten zur Konvertierung von C++-Anwendungen.
3. Wählen Sie unter Compilertyp den Compiler aus, den Sie für den Quellcode Ihrer C++-Anwendung verwenden. AWS SCT unterstützt die folgenden C++-Compiler: Microsoft Visual C++, GCC, die GNU Compiler Collection und Clang. Die Standardoption ist Microsoft Visual C++.
4. Geben Sie für Benutzerdefinierte Makros den Pfad zu der Datei ein, die benutzerdefinierte Makros aus Ihrem C++-Projekt enthält. Stellen Sie sicher, dass diese Datei die folgende Struktur hat:. `#define name value` Im vorherigen Beispiel `value` ist dies ein optionaler Parameter. Der Standardwert für diesen optionalen Parameter ist1.

   Um diese Datei zu erstellen, öffnen Sie Ihr Projekt in Microsoft Visual Studio und wählen Sie dann Project, Properties, C/C++ und Preprocessor aus. Wählen Sie für Präprozessor-Definitionen die Option Bearbeiten und kopieren Sie Namen und Werte in eine neue Textdatei. Fügen Sie dann für jede Zeichenfolge in der Datei das folgende Präfix hinzu:`#define` .
5. Geben Sie für Externe Include-Verzeichnisse die Pfade zu den Ordnern ein, die externe Bibliotheken enthalten, die Sie in Ihrem C++-Projekt verwenden.
6. Wählen Sie im linken Bereich die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
7. Wählen Sie „Konvertieren“. AWS SCT analysiert Ihre Quellcodedateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen C++-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

AWS SCT Erstellt im Bereich Zieldatenbank eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen, wie im Folgenden gezeigt.



8. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter Speichern Sie Ihren konvertierten Anwendungscode.

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Gehen Sie wie folgt vor, um Ihren konvertierten Anwendungscode zu speichern.

Um den konvertierten Anwendungscode zu speichern

1. Erweitern Sie den C++-Knoten unter Anwendungen im Bereich Zieldatenbank.
2. Wählen Sie Ihre konvertierte Anwendung aus und klicken Sie auf Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Ordner auswählen.

Verwaltung von Projekten zur Konvertierung von C++-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von C++-Anwendungen hinzufügen, Konvertierungseinstellungen bearbeiten, den C++-Anwendungscode aktualisieren oder ein C++-Konvertierungsprojekt aus Ihrem AWS SCT Projekt entfernen.

Um ein zusätzliches Projekt zur Konvertierung von C++-Anwendungen hinzuzufügen

1. Erweitern Sie den Knoten Anwendungen im linken Bereich.
2. Wählen Sie den C++-Knoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.
4. Geben Sie die Informationen ein, die zum Erstellen eines neuen Projekts zur Konvertierung von C++-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter Projekte zur Konvertierung von C++-Anwendungen erstellen.

Sie können Konvertierungseinstellungen für alle Projekte zur Konvertierung von C++-Anwendungen in Ihrem AWS SCT Projekt angeben.

Um die Konvertierungseinstellungen für alle C++-Anwendungen zu bearbeiten

1. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen und dann Anwendungskonvertierung aus.
2. Wählen Sie unter Compilertyp den Compiler aus, den Sie für den Quellcode Ihrer C++-Anwendung verwenden. AWS SCT unterstützt die folgenden C++-Compiler: Microsoft Visual C++, GCC, die GNU Compiler Collection und Clang. Die Standardoption ist Microsoft Visual C++.
3. Geben Sie für Benutzerdefinierte Makros den Pfad zu der Datei ein, die benutzerdefinierte Makros aus Ihrem C++-Projekt enthält. Stellen Sie sicher, dass diese Datei die folgende Struktur hat:. `#define name value` Im vorherigen Beispiel `value` ist dies ein optionaler Parameter. Der Standardwert für diesen optionalen Parameter ist1.

   Um diese Datei zu erstellen, öffnen Sie Ihr Projekt in Microsoft Visual Studio und wählen Sie dann Project, Properties, C/C++ und Preprocessor aus. Wählen Sie für Präprozessor-Definitionen die Option Bearbeiten und kopieren Sie Namen und Werte in eine neue Textdatei. Fügen Sie dann für jede Zeichenfolge in der Datei das folgende Präfix hinzu:`#define` .

4. Geben Sie für Externe Include-Verzeichnisse die Pfade zu den Ordnern ein, die externe Bibliotheken enthalten, die Sie in Ihrem C++-Projekt verwenden.
5. Wählen Sie OK, um die Projekteinstellungen zu speichern und das Fenster zu schließen.

Sie können auch Konvertierungseinstellungen für jedes Projekt zur Konvertierung von C++-Anwendungen angeben. Weitere Informationen finden Sie unter Konvertieren Sie den SQL-Code Ihrer C++-Anwendung.

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in das AWS SCT Projekt hoch.

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie den C++-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu aktualisierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Aktualisieren“ und dann „Ja“.

   AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um die Codeänderungen, die Sie in vorgenommen haben, AWS SCT und die Konvertierungsergebnisse beizubehalten, erstellen Sie ein neues C++-Konvertierungsprojekt.

   AWS SCT Entfernt außerdem die Einstellungen für die Anwendungskonvertierung, die Sie für die ausgewählte Anwendung angegeben haben. AWS SCT Wendet nach dem Hochladen des aktualisierten Anwendungscodes die Standardwerte aus den Projekteinstellungen an.

Um ein Projekt zur Konvertierung einer C++-Anwendung zu entfernen

1. Erweitern Sie den C++-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu entfernende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Löschen und dann OK.

Erstellen Sie einen Bewertungsbericht zur Konvertierung von C++-Anwendungen in AWS SCT

Der Bewertungsbericht zur Konvertierung von C++-Anwendungen enthält Informationen zur Konvertierung des in Ihrer C++-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, der nicht konvertiert AWS SCT werden kann.

Um einen Bewertungsbericht zur Konvertierung von C++-Anwendungen zu erstellen

1. Erweitern Sie im linken Bereich den C++-Knoten unter Anwendungen.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“.
4. Wählen Sie im Menü „Ansicht“ die Option „Bewertungsbericht anzeigen“.
5. Rufen Sie die Registerkarte „Zusammenfassung“ auf.

   Auf der Registerkarte Zusammenfassung werden die zusammenfassenden Informationen aus dem Bericht zur Bewertung der C++-Anwendung angezeigt. Es werden Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien angezeigt.
6. Wählen Sie Anweisungen in JSON speichern, um den extrahierten SQL-Code aus Ihrer Java-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Wählen Sie oben rechts die Option Als PDF speichern, um den Bericht als PDF-Datei zu speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.
   - Wählen Sie oben rechts die Option In CSV speichern, um den Bericht als CSV-Datei zu speichern.

     Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertieren von SQL-Code in Java-Anwendungen mit AWS Schema Conversion Tool

Für eine Konvertierung von Oracle nach PostgreSQL können AWS Schema Conversion Tool Sie den in Ihre Java-Anwendungen eingebetteten SQL-Code konvertieren. Dieser spezielle Java-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, z. B. Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für Java-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Wenn Ihre Java-Anwendung das MyBatis Framework für die Interaktion mit Datenbanken verwendet, können Sie es verwenden, AWS SCT um in MyBatis XML-Dateien und Anmerkungen eingebettete SQL-Anweisungen zu konvertieren. Um die Logik dieser SQL-Anweisungen zu verstehen, AWS SCT verwendet die MyBatis Konfigurationsdatei. AWS SCT kann diese Datei automatisch in Ihrem Anwendungsordner finden, oder Sie können den Pfad zu dieser Datei manuell eingeben.

Projekte zur Konvertierung von Java-Anwendungen erstellen in AWS SCT

Sie können ein Konvertierungsprojekt für Java-Anwendungen nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.

Sie können mehrere Projekte zur Anwendungskonvertierung in einem einzigen AWS SCT Projekt hinzufügen. Gehen Sie wie folgt vor, um ein Projekt zur Konvertierung von Java-Anwendungen zu erstellen.

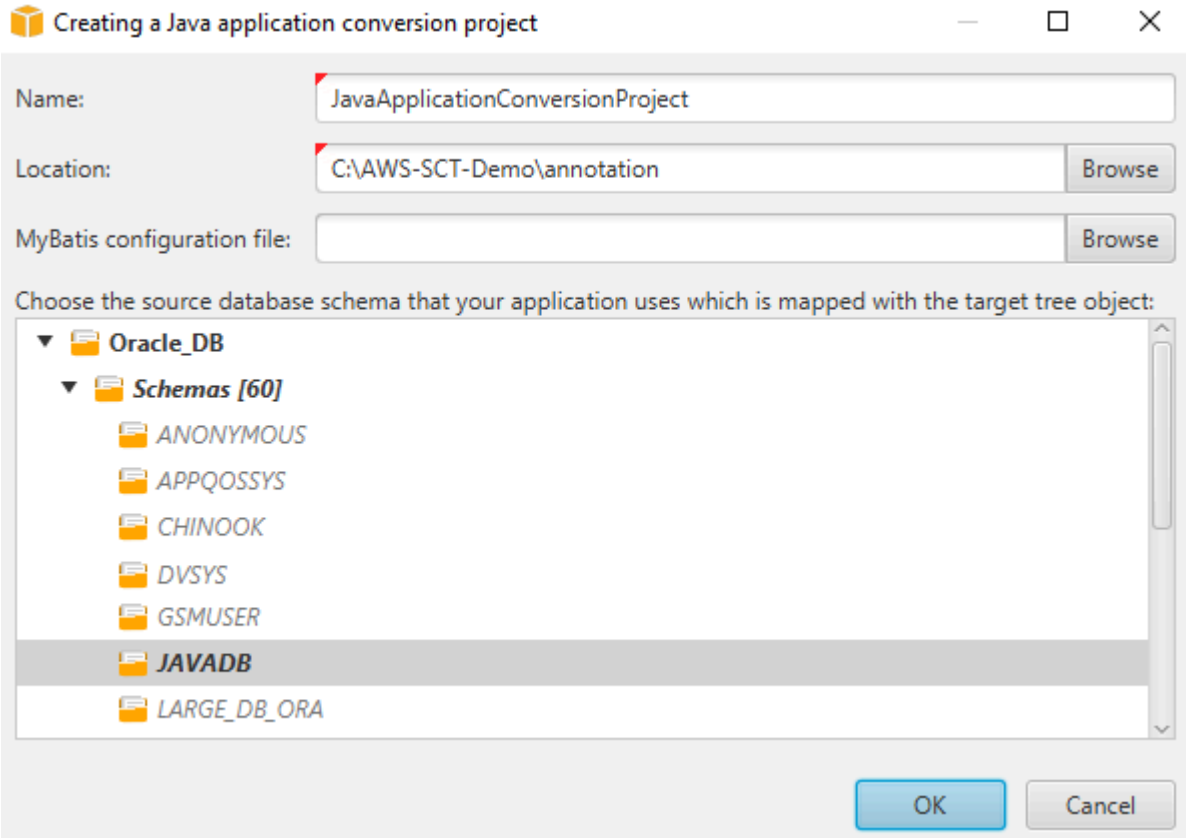
Um ein Projekt zur Konvertierung von Java-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter Projekte starten und verwalten in AWS SCT und Server zum Projekt hinzufügen in AWS SCT.
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere

Informationen erhalten Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool und Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.

3. Wählen Sie im Menü Ansicht die Option Hauptansicht.
4. Wählen Sie im Menü Anwendungen die Option Neue Java-Anwendung aus.

   Das Dialogfeld „Konvertierungsprojekt für Java-Anwendungen erstellen“ wird angezeigt.



5. Geben Sie unter Name einen Namen für Ihr Projekt zur Konvertierung von Java-Anwendungen ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der sinnvoll ist, wenn Sie mehrere Projekte hinzufügen.
6. Geben Sie unter Standort den Speicherort des Quellcodes für Ihre Anwendung ein.
7. (Optional) Geben Sie für die MyBatis Konfigurationsdatei den Pfad zur MyBatis Konfigurationsdatei ein. AWS SCT scannt Ihren Anwendungsordner, um diese Datei automatisch zu erkennen. Wenn sich diese Datei nicht in Ihrem Anwendungsordner befindet oder wenn Sie mehrere Konfigurationsdateien verwenden, geben Sie den Pfad manuell ein.
8. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.

9. Wählen Sie OK, um Ihr Projekt zur Konvertierung von Java-Anwendungen zu erstellen.
10. Suchen Sie Ihr Projekt zur Konvertierung von Java-Anwendungen im linken Bereich im Knoten Anwendungen.

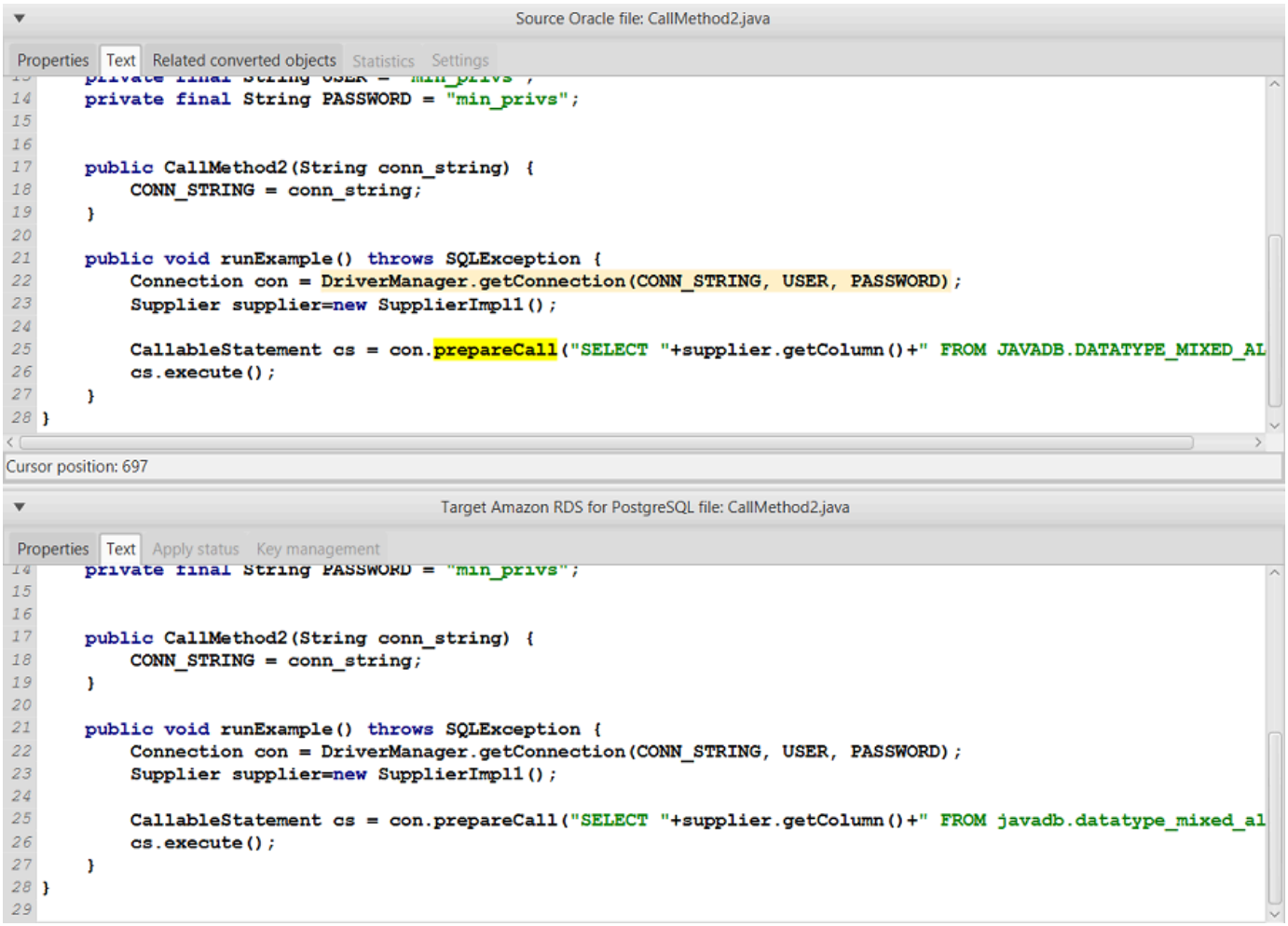
Konvertieren Sie den SQL-Code Ihrer Java-Anwendung in AWS SCT

Nachdem Sie Ihre Java-Anwendung zum AWS SCT Projekt hinzugefügt haben, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre Java-Anwendung eingebetteten SQL-Code in der zu analysieren und zu konvertieren AWS Schema Conversion Tool.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie den Java-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“. AWS SCT analysiert Ihre Quellcodedateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen Java-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

   AWS SCT Erstellt im Bereich Zieldatenbank eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.



4. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter Speichern Sie Ihren konvertierten Anwendungscode.

Ihre Java-Anwendungen können SQL-Code enthalten, der mit verschiedenen Quelldatenbanken interagiert. Sie können mehrere dieser Quelldatenbanken zu PostgreSQL migrieren. Stellen Sie in diesem Fall sicher, dass Sie keinen SQL-Code konvertieren, der mit Datenbanken interagiert, die Sie aus dem Migrationsbereich ausgeschlossen haben. Sie können Quelldateien Ihrer Java-Anwendung vom Konvertierungsbereich ausschließen. Deaktivieren Sie dazu die Kontrollkästchen für die Namen der Dateien, die Sie aus dem Konvertierungsbereich ausschließen möchten.

Nachdem Sie den Konvertierungsbereich geändert haben, analysiert es AWS SCT weiterhin den SQL-Code aller Quelldateien Ihrer Java-Anwendungen. AWS SCT Kopiert dann alle Quelldateien, die Sie aus dem Konvertierungsbereich ausgeschlossen haben, in den Zielordner. Dieser Vorgang ermöglicht es, Ihre Anwendung zu erstellen, nachdem Sie Ihre konvertierten Anwendungsdateien gespeichert haben.

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Gehen Sie wie folgt vor, um den konvertierten Anwendungscode zu speichern.

Um den konvertierten Anwendungscode zu speichern

1. Erweitern Sie den Java-Knoten unter Anwendungen im Bereich Zieldatenbank.
2. Wählen Sie Ihre konvertierte Anwendung aus und klicken Sie auf Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Ordner auswählen.

Wenn Ihre Java-Quellanwendung das MyBatis Framework verwendet, stellen Sie sicher, dass Sie Ihre Konfigurationsdatei aktualisieren, damit sie mit Ihrer neuen Datenbank funktioniert.

Verwaltung von Projekten zur Konvertierung von Java-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von Java-Anwendungen hinzufügen, den Anwendungscode im AWS SCT Projekt aktualisieren oder ein Java-Konvertierungsprojekt aus Ihrem AWS SCT Projekt entfernen.

Um ein zusätzliches Projekt zur Konvertierung von Java-Anwendungen hinzuzufügen

1. Erweitern Sie den Knoten Anwendungen im linken Bereich.
2. Wählen Sie den Java-Knoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.
4. Geben Sie die Informationen ein, die zum Erstellen eines neuen Projekts zur Konvertierung von Java-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter Projekte zur Konvertierung von Java-Anwendungen erstellen.

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in das AWS SCT Projekt hoch.

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie den Java-Knoten unter Anwendungen im linken Bereich.

2. Wählen Sie die zu aktualisierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Aktualisieren“ und dann „Ja“.

   AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um die von Ihnen vorgenommenen Codeänderungen AWS SCT und die Konvertierungsergebnisse beizubehalten, erstellen Sie ein neues Java-Konvertierungsprojekt.

Wenn Ihre Java-Quellanwendung das MyBatis Framework verwendet, AWS SCT verwendet die MyBatis Konfigurationsdatei, um Ihren SQL-Code zu analysieren. Nachdem Sie diese Datei geändert haben, laden Sie sie in das AWS SCT Projekt hoch.

Um den Pfad zur MyBatis Konfigurationsdatei zu bearbeiten

1. Erweitern Sie den Java-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie Ihre Anwendung und dann Einstellungen aus.
3. Wählen Sie Durchsuchen und wählen Sie dann die MyBatis Konfigurationsdatei aus.
4. Wählen Sie Anwenden aus.
5. Wählen Sie im linken Bereich Ihre Anwendung aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie „Aktualisieren“.

Um ein Konvertierungsprojekt für Java-Anwendungen zu entfernen

1. Erweitern Sie den Java-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu entfernende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Löschen und dann OK.

Erstellen Sie einen Bewertungsbericht zur Konvertierung von Java-Anwendungen in AWS SCT

Der Bewertungsbericht zur Konvertierung von Java-Anwendungen enthält Informationen zur Konvertierung des in Ihre Java-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-

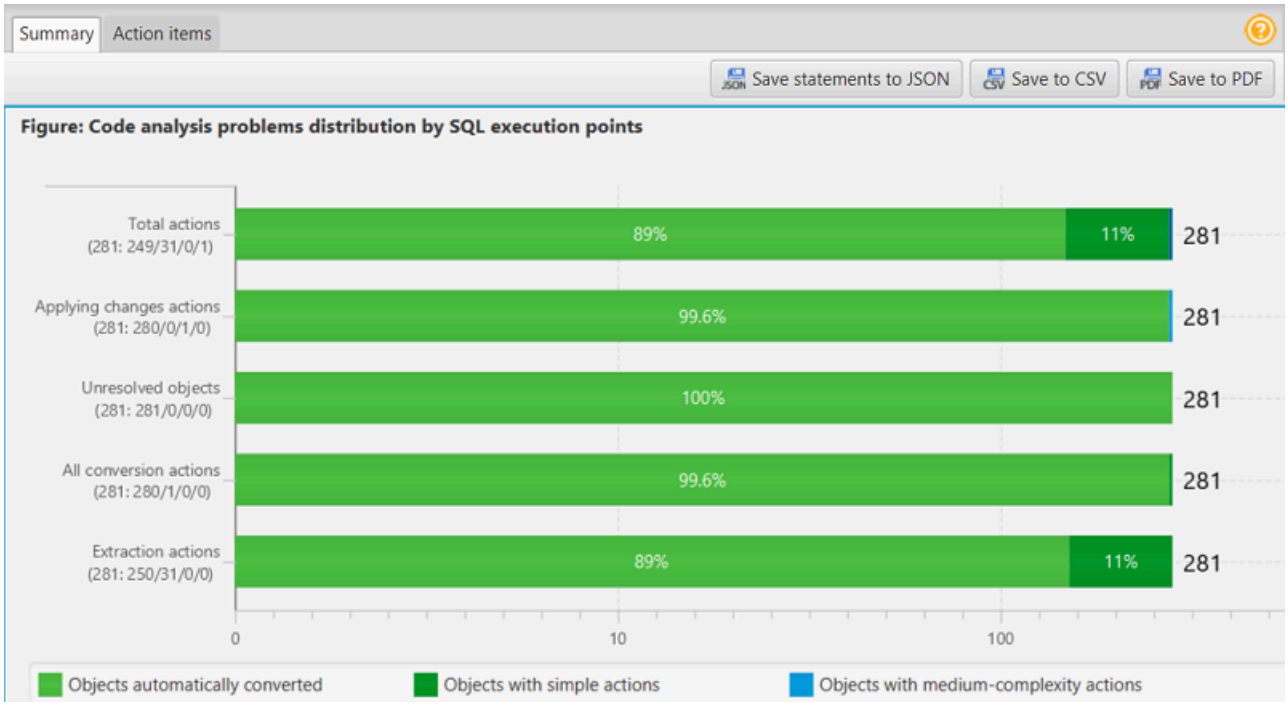
Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, der nicht konvertiert AWS SCT werden kann.

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Konvertierung von Java-Anwendungen zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Konvertierung von Java-Anwendungen

1. Erweitern Sie im linken Bereich den Java-Knoten unter Anwendungen.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“.
4. Wählen Sie im Menü „Ansicht“ die Option „Bewertungsbericht anzeigen“.
5. Prüfen Sie die Registerkarte Summary.

   Auf der Registerkarte „Zusammenfassung“ (siehe unten) werden die zusammenfassenden Informationen aus dem Bewertungsbericht für die Java-Anwendung angezeigt. Es werden Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien angezeigt.



6. Wählen Sie Anweisungen in JSON speichern, um den extrahierten SQL-Code aus Ihrer Java-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Wählen Sie oben rechts die Option Als PDF speichern, um den Bericht als PDF-Datei zu speichern.

  Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.
- Wählen Sie oben rechts die Option In CSV speichern, um den Bericht als CSV-Datei zu speichern.

  Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertierung von SQL-Code in Pro*C-Anwendungen mit AWS Schema Conversion Tool

Für eine Konvertierung von Oracle nach PostgreSQL können Sie die AWS Schema Conversion Tool (AWS SCT) verwenden, um in Ihre Pro*C-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle Pro*C-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, wie Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für Pro*C-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Projekte zur Konvertierung von Pro*C-Anwendungen erstellen in AWS SCT

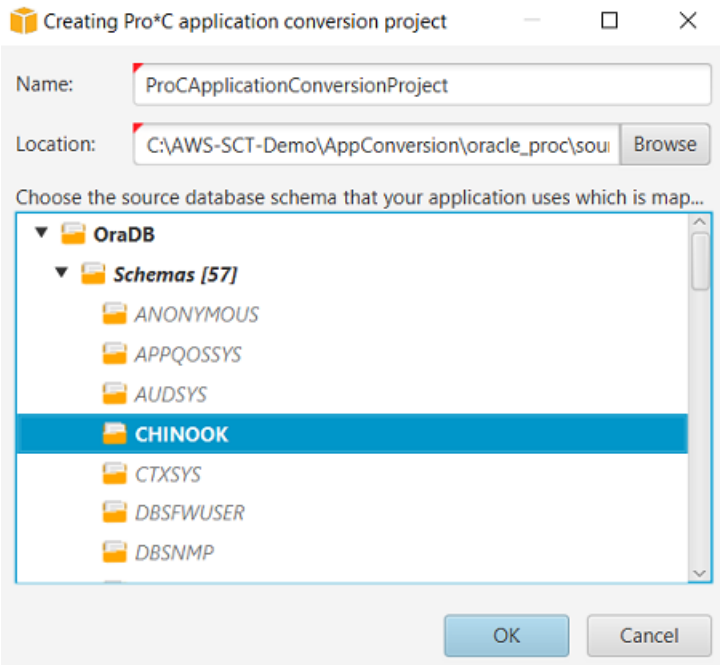
Sie können ein Pro*C-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool.

Sie können mehrere Projekte zur Anwendungskonvertierung in einem einzigen AWS SCT Projekt hinzufügen. Gehen Sie wie folgt vor, um ein Projekt zur Konvertierung von Pro*C-Anwendungen zu erstellen.

Um ein Projekt zur Konvertierung von Pro*C-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter Projekte starten und verwalten in AWS SCT und Server zum Projekt hinzufügen in AWS SCT.
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter Zuordnung von Datentypen in der AWS Schema Conversion Tool und Zuordnung zu virtuellen Zielen im AWS Schema Conversion Tool.
3. Wählen Sie im Menü Ansicht die Option Hauptansicht.
4. Wählen Sie im Menü „Anwendungen“ die Option „Neue Pro*C-Anwendung“.

   Das Dialogfeld „Konvertierungsprojekt für eine Pro*C-Anwendung erstellen“ wird angezeigt.



5. Geben Sie unter Name einen Namen für Ihr Projekt zur Konvertierung Ihrer Pro*C-Anwendung ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Geben Sie unter Standort den Speicherort des Quellcodes für Ihre Anwendung ein.
7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.

8. Wählen Sie OK, um Ihr Projekt zur Konvertierung Ihrer Pro*C-Anwendung zu erstellen.
9. Suchen Sie Ihr Projekt zur Konvertierung von Pro*C-Anwendungen im linken Bereich im Knoten Anwendungen.

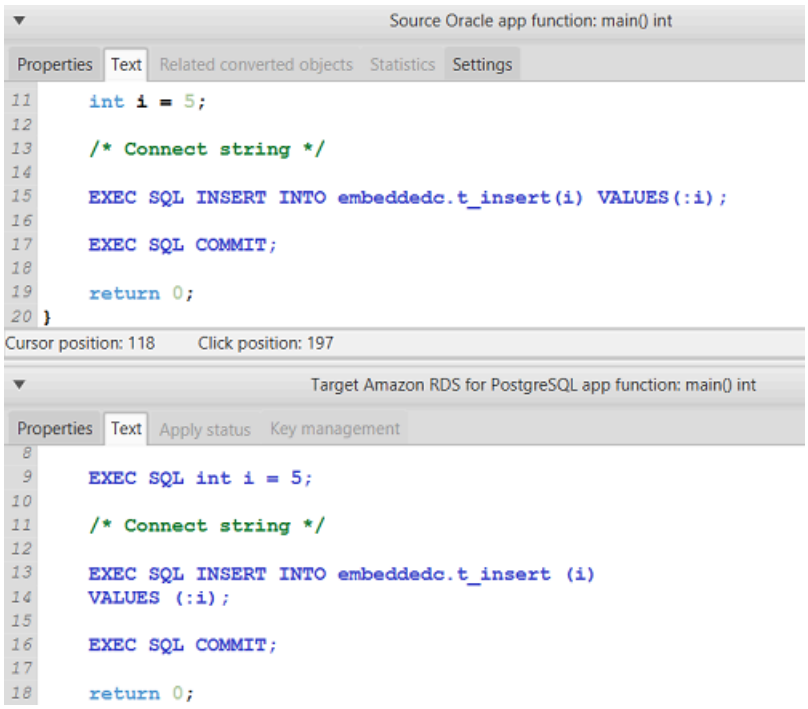
Konvertieren Sie den SQL-Code Ihrer Pro*C-Anwendung in AWS SCT

Nachdem Sie Ihre Pro*C-Anwendung zum AWS SCT Projekt hinzugefügt haben, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre Pro*C-Anwendung eingebetteten SQL-Code in der zu analysieren und zu konvertieren. AWS Schema Conversion Tool

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie den Pro*C-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu konvertierende Anwendung und anschließend Einstellungen aus.
    a. Geben Sie unter Globaler Header-Dateipfad den Pfad zu den Header-Dateien ein, die Ihr Anwendungsprojekt verwendet.
    b. Wählen Sie Alle ungelösten Hostvariablen interpretieren, um alle ungelösten Variablen im konvertierten Code zu sehen.
    c. Wählen Sie „Konvertierungsfunktion für Zeichenketten mit fester Breite aus dem Erweiterungspaket verwenden“, um die Funktionen des Erweiterungspakets im konvertierten SQL-Code zu verwenden. AWS SCT schließt die Erweiterungspaketdateien in Ihr Anwendungsprojekt ein.
    d. Wählen Sie Anonyme PL/SQL-Blöcke in eigenständige SQL-Aufrufe oder gespeicherte Funktionen umwandeln, um in Ihrer Zieldatenbank gespeicherte Prozeduren für alle anonymen PL/SQL-Blöcke zu erstellen. AWS SCT schließt dann die Ausführungen dieser gespeicherten Prozeduren in den konvertierten Anwendungscode ein.
    e. Wählen Sie Benutzerdefinierten Cursorfluss verwenden, um die Konvertierung von Oracle-Datenbankcursorn zu verbessern.
3. Wählen Sie im linken Bereich die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
4. Wählen Sie „Konvertieren“. AWS SCT analysiert Ihre Quellcodedateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen Pro*C-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

AWS SCT Erstellt im Bereich Zieldatenbank eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.



5. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter Bearbeiten und Speichern Ihres konvertierten Anwendungscodes.

Bearbeiten und speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Sie können die konvertierten SQL-Anweisungen bearbeiten und verwenden AWS SCT , um diesen bearbeiteten Code in den konvertierten Pro*C-Anwendungscode einzubetten. Gehen Sie wie folgt vor, um Ihren konvertierten SQL-Code zu bearbeiten.

Um Ihren konvertierten SQL-Code zu bearbeiten

1. Erweitern Sie den Pro*C-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu konvertierende Anwendung aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie „Konvertieren“.
3. Wählen Sie im Menü „Ansicht“ die Option „Bewertungsbericht anzeigen“.
4. Wählen Sie Anweisungen in CSV speichern, um den extrahierten SQL-Code aus Ihrer Pro*C-Anwendung als CSV-Datei zu speichern.

5. Geben Sie den Namen der CSV-Datei ein, um den extrahierten SQL-Code zu speichern, und wählen Sie Speichern.
6. Bearbeiten Sie den extrahierten SQL-Code.
7. Wählen Sie im Menü Ansicht die Option Hauptansicht.
8. Erweitern Sie den Knoten Pro*C unter Anwendungen im Bereich Zieldatenbank.
9. Wählen Sie Ihre konvertierte Anwendung aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Anweisungen aus CSV importieren.
10. Wählen Sie Ja, wählen Sie dann die Datei mit Ihrem bearbeiteten SQL-Code aus und wählen Sie Öffnen.

AWS SCT zerlegt die konvertierten SQL-Anweisungen in Teile und platziert sie in den entsprechenden Objekten Ihres Quellanwendungscodes. Gehen Sie wie folgt vor, um den konvertierten Anwendungscode zu speichern.

Um den konvertierten Anwendungscode zu speichern

1. Erweitern Sie den Knoten Pro*C unter Anwendungen im Bereich Zieldatenbank.
2. Wählen Sie Ihre konvertierte Anwendung aus und klicken Sie auf Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Ordner auswählen.

Projekte zur Konvertierung von Pro*C-Anwendungen verwalten in AWS SCT

Sie können mehrere Projekte zur Konvertierung von Pro*C-Anwendungen hinzufügen, den Anwendungscode im Projekt aktualisieren oder ein AWS SCT Pro*C-Konvertierungsprojekt aus Ihrem Projekt entfernen. AWS SCT

Um ein zusätzliches Projekt zur Konvertierung von Pro*C-Anwendungen hinzuzufügen

1. Erweitern Sie den Knoten Anwendungen im linken Bereich.
2. Wählen Sie den Knoten Pro*C und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.

4. Geben Sie die Informationen ein, die für die Erstellung eines neuen Projekts zur Konvertierung von Pro*C-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter Projekte zur Konvertierung von Pro*C-Anwendungen erstellen.

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in das AWS SCT Projekt hoch.

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie im linken Bereich den Knoten Pro*C unter Anwendungen.
2. Wählen Sie die zu aktualisierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Aktualisieren“ und dann „Ja“.

   AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um die Codeänderungen, die Sie in vorgenommen haben, AWS SCT und die Konvertierungsergebnisse beizubehalten, erstellen Sie ein neues Pro*C-Konvertierungsprojekt.

Um ein Projekt zur Konvertierung einer Pro*C-Anwendung zu entfernen

1. Erweitern Sie im linken Bereich den Knoten Pro*C unter Anwendungen.
2. Wählen Sie die zu entfernende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Löschen und dann OK.

Erstellen Sie einen Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen in AWS SCT

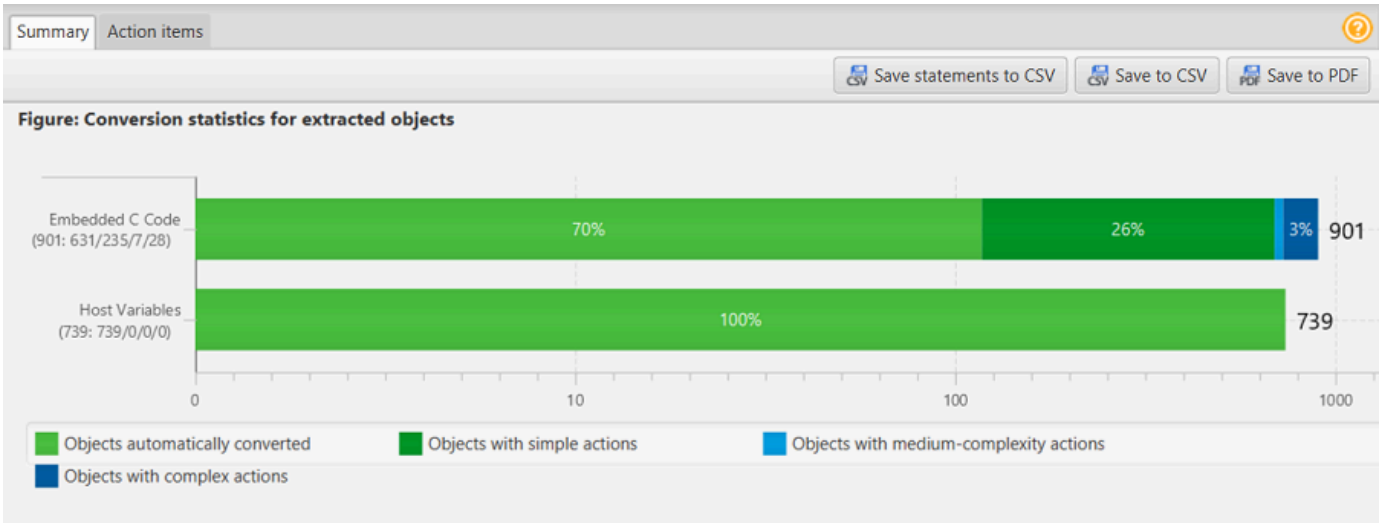
Der Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen enthält Informationen zur Konvertierung des in Ihrer Pro*C-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, der nicht konvertiert AWS SCT werden kann.

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen

1. Erweitern Sie im linken Bereich den Knoten Pro*C unter Anwendungen.
2. Wählen Sie die zu konvertierende Anwendung aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie „Konvertieren“.
4. Wählen Sie im Menü „Ansicht“ die Option „Bewertungsbericht anzeigen“.
5. Prüfen Sie die Registerkarte Summary.

   Auf der Registerkarte „Zusammenfassung“ (siehe unten) werden die zusammenfassenden Informationen aus dem Bewertungsbericht für die Pro*C-Anwendung angezeigt. Es zeigt Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien.



6. Wählen Sie Anweisungen in CSV speichern, um den extrahierten SQL-Code aus Ihrer Pro*C-Anwendung als Datei mit kommagetrennten Werten (CSV) zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

   - Wählen Sie oben rechts die Option Als PDF speichern, um den Bericht als PDF-Datei zu speichern.

     Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.
   - Wählen Sie oben rechts die Option In CSV speichern, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Verwenden von Erweiterungspaketen mit AWS Schema Conversion Tool

Ein AWS SCT Erweiterungspaket ist ein Zusatzmodul, das Funktionen emuliert, die in einer Quelldatenbank vorhanden sind und für die Konvertierung von Objekten in die Zieldatenbank erforderlich sind. Bevor Sie ein AWS SCT Erweiterungspaket installieren können, konvertieren Sie Ihr Datenbankschema.

Jedes AWS SCT Erweiterungspaket umfasst die folgenden Komponenten:

- DB-Schema — Beinhaltet SQL-Funktionen, -Prozeduren und -Tabellen zur Emulation bestimmter Datenbankobjekte wie Sequenzen im Bereich Online-Transaktionsverarbeitung (OLTP) und Online-Analytische Verarbeitung (OLAP). Emuliert außerdem nicht von der Quelldatenbank aus. built-in-functions Der Name dieses Schemas hat das folgende Format:. `aws_`*`database_engine_name`*`_ext`
- AWS Lambda Funktionen (für bestimmte OLTP-Datenbanken) — Beinhaltet AWS Lambda Funktionen, die komplexe Datenbankfunktionen emulieren, z. B. die Auftragsplanung und das Senden von E-Mails.
- Benutzerdefinierte Bibliotheken für OLAP-Datenbanken — Enthält eine Reihe von Java- und Python-Bibliotheken, mit denen Sie Microsoft SQL Server Integration Services (SSIS) -Skripts zum Extrahieren, Transformieren und Laden (ETL) nach AWS Glue oder migrieren können. AWS Glue Studio

  Java-Bibliotheken umfassen die folgenden Module:

  - `spark-excel_2.11-0.13.1.jar`— Um die Funktionalität von Excel-Quell- und Zielkomponenten zu emulieren.
  - `spark-xml_2.11-0.9.0.jarpoi-ooxml-schemas-4.1.2.jar`, und `xmlbeans-3.1.0.jar` — Um die Funktionalität der XML-Quellkomponente zu emulieren.

  Python-Bibliotheken umfassen die folgenden Module:

  - `sct_utils.py`— Um Quelldatentypen zu emulieren und Parameter für die Spark-SQL-Abfrage vorzubereiten.
  - `ssis_datetime.py`— Um integrierte Funktionen für Datum und Uhrzeit zu emulieren.
  - `ssis_null.py`— Um die `ISNULL` und die `REPLACENULL` integrierten Funktionen zu emulieren.
  - `ssis_string.py`— Um eingebaute Zeichenkettenfunktionen zu emulieren.

Weitere Informationen zu diesen Bibliotheken finden Sie unter Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete.

Sie können AWS SCT Erweiterungspakete auf zwei Arten anwenden:

- AWS SCT kann ein Erweiterungspaket automatisch anwenden, wenn Sie ein Zieldatenbankskript anwenden, indem Sie im Kontextmenü Auf Datenbank anwenden wählen. AWS SCT wendet das Erweiterungspaket an, bevor es alle anderen Schemaobjekte anwendet.
- Um ein Erweiterungspaket manuell anzuwenden, wählen Sie die Zieldatenbank aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Erweiterungspaket anwenden für. In den meisten Situationen ist eine automatische Anwendung ausreichend. Möglicherweise möchten Sie das Paket jedoch manuell anwenden, falls es versehentlich gelöscht wurde.

Jedes Mal, wenn Sie ein AWS SCT Erweiterungspaket auf einen Zieldatenspeicher anwenden, werden die Komponenten überschrieben, und es AWS SCT wird eine entsprechende Benachrichtigung angezeigt. Um diese Benachrichtigungen zu deaktivieren, wählen Sie Einstellungen, Allgemeine Einstellungen, Benachrichtigungen und dann die Option Warnung zum Austausch des Erweiterungspakets ausblenden aus.

Für eine Konvertierung von Microsoft SQL Server zu PostgreSQL können Sie das SQL Server to PostgreSQL Extension Pack in verwenden. AWS SCT Dieses Erweiterungspaket emuliert SQL Server Agent und SQL Server Database Mail. Weitere Informationen erhalten Sie unter Emulieren des SQL Server-Agents in PostgreSQL mit einem Erweiterungspaket und Emulieren von SQL Server-Datenbank-Mail in PostgreSQL mit einem Erweiterungspaket.

Im Folgenden finden Sie weitere Informationen zum Arbeiten mit AWS SCT Erweiterungspaketen.

Themen

- Berechtigungen für die Verwendung des AWS SCT Erweiterungspakets
- Verwenden Sie das Erweiterungspaket-Schema
- Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete
- Verwenden der AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket
- Funktionen für das AWS SCT Erweiterungspaket konfigurieren

Berechtigungen für die Verwendung des AWS SCT Erweiterungspakets

Das AWS SCT Erweiterungspaket für Amazon Aurora emuliert das Senden von E-Mails, die Auftragsplanung, das Einstellen von Warteschlangen und andere Operationen mithilfe von AWS Lambda Funktionen. Wenn Sie das AWS SCT Erweiterungspaket auf Ihre Aurora-Zieldatenbank anwenden, AWS SCT erstellt es eine neue AWS Identity and Access Management (IAM-) Rolle und eine Inline-IAM-Richtlinie. Als Nächstes AWS SCT erstellt es eine neue Lambda-Funktion und konfiguriert Ihren Aurora-DB-Cluster für ausgehende Verbindungen zu. AWS Lambda Um diese Operationen auszuführen, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die folgenden erforderlichen Berechtigungen gewähren:

- `iam:CreateRole`— um eine neue IAM-Rolle für Ihr AWS Konto zu erstellen.
- `iam:CreatePolicy`— um eine neue IAM-Richtlinie für Ihr AWS Konto zu erstellen.
- `iam:AttachRolePolicy`— um die angegebene Richtlinie an Ihre IAM-Rolle anzuhängen.
- `iam:PutRolePolicy`— um ein integriertes Richtliniendokument zu aktualisieren, das in Ihre IAM-Rolle eingebettet ist.
- `iam:PassRole`— um die angegebene IAM-Rolle an die Regel-Engine zu übergeben.
- `iam:TagRole`— um einer IAM-Rolle Tags hinzuzufügen.
- `iam:TagPolicy`— um einer IAM-Richtlinie Tags hinzuzufügen.
- `lambda:ListFunctions`— um die Liste Ihrer Lambda-Funktionen zu sehen.
- `lambda:ListTags`— um die Liste der Tags Ihrer Lambda-Funktionen zu sehen.
- `lambda:CreateFunction`— um eine neue Lambda-Funktion zu erstellen.
- `rds:AddRoleToDBCluster`— um Ihrem Aurora-DB-Cluster eine IAM-Rolle zuzuordnen.

Das AWS SCT Erweiterungspaket für Amazon Redshift emuliert Quell-Data Warehouse-Basisfunktionen, die erforderlich sind, wenn konvertierte Objekte auf Amazon Redshift angewendet werden. Bevor Sie Ihren konvertierten Code auf Amazon Redshift anwenden, müssen Sie das Erweiterungspaket für Amazon Redshift anwenden. Nehmen Sie dazu die `iam:SimulatePrincipalPolicy` Aktion in Ihre IAM-Richtlinie auf.

AWS SCT verwendet den IAM Policy Simulator, um die erforderlichen Berechtigungen für die Installation des Amazon Redshift Redshift-Erweiterungspakets zu überprüfen. Der IAM Policy Simulator kann eine Fehlermeldung anzeigen, auch wenn Sie Ihren IAM-Benutzer korrekt konfiguriert

haben. Dies ist ein bekanntes Problem des IAM Policy Simulator. Außerdem zeigt der IAM-Richtliniensimulator eine Fehlermeldung an, wenn Sie die `iam:SimulatePrincipalPolicy` Aktion nicht in Ihrer IAM-Richtlinie haben. In diesen Fällen können Sie die Fehlermeldung ignorieren und das Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anwenden. Weitere Informationen finden Sie unter Anwenden des Erweiterungspakets.

Verwenden Sie das Erweiterungspaket-Schema

Wenn Sie Ihr Datenbank- oder Data Warehouse-Schema konvertieren, fügt AWS SCT ein zusätzliches Schema zur Zieldatenbank hinzu. Über dieses Schema werden SQL-Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Zieldatenbank benötigt werden. Dieses Zusätzliche Schema wird als Erweiterungspaketschema bezeichnet.

Das Erweiterungspaketschema für OLTP-Datenbanken wird wie folgt entsprechend der Quelldatenbank benannt:

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

Das Erweiterungspaketschema für OLAP-Datawarehouse-Anwendungen wird wie folgt entsprechend dem Quelldatenspeicher benannt:

- 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`

Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete

In einigen Fällen AWS SCT können Quelldatenbank-Features nicht in äquivalente Features in Ihrer Zieldatenbank konvertiert werden. Das entsprechende AWS SCT Erweiterungspaket enthält benutzerdefinierte Bibliotheken, die einige Funktionen der Quelldatenbank in Ihrer Zieldatenbank emulieren.

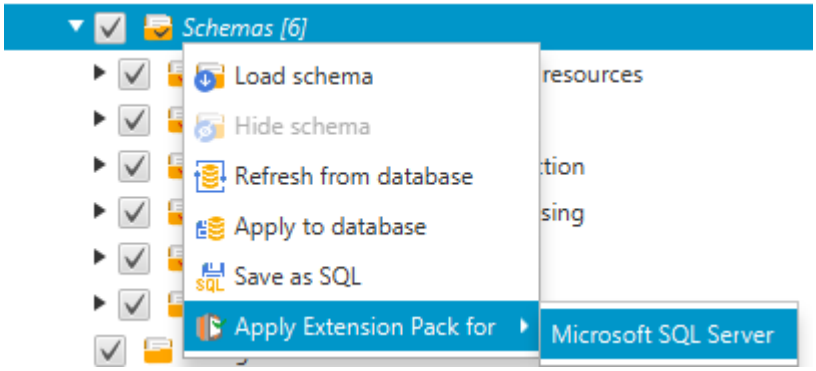
Informationen zum Konvertieren einer Transaktionsdatenbank finden Sie unter. Verwenden der AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket

Anwenden des Erweiterungspakets

Sie können das AWS SCT Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anwenden oder wenn Sie den konvertierten Code auf Ihre Zieldatenbank anwenden.

Um das Erweiterungspaket mit dem Erweiterungspaket-Assistenten anzuwenden

1. Öffnen Sie in der Zieldatenbankstruktur das Kontextmenü (Rechtsklick), wählen Sie Erweiterungspaket anwenden für und wählen Sie dann Ihre Quelldatenbankplattform aus. AWS Schema Conversion Tool



   Der Assistent des Erweiterungspakets wird angezeigt.
2. Lesen Sie die Seite Welcome und wählen Sie dann Next aus.
3. Gehen Sie auf der Seite mit den AWS Profileinstellungen wie folgt vor:

   - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus. Die Option „Diesen Schritt vorerst überspringen“ ist nur für OLTP-Datenbanken (Online Transaction Processing) verfügbar.

- Wenn Sie eine neue Bibliothek hochladen, geben Sie die Anmeldeinformationen ein, um eine Verbindung zu Ihrer Bibliothek herzustellen. AWS-Konto Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Sie können Ihre AWS Command Line Interface (AWS CLI) -Anmeldeinformationen verwenden, wenn Sie die AWS CLI installiert haben. Sie können auch die Anmeldeinformationen verwenden, die Sie zuvor in einem Profil in den globalen Anwendungseinstellungen gespeichert und mit dem Projekt verknüpft haben. Wählen Sie bei Bedarf zu globalen Einstellungen navigieren, um Ihr AWS SCT Projekt zu konfigurieren oder einem anderen Profil zuzuordnen. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.

4. Wenn Sie eine neue Bibliothek hochladen, wählen Sie auf der Seite zum Hochladen der Bibliothek die Option Ich muss eine Bibliothek hochladen. Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Geben Sie als Nächstes den Amazon S3 S3-Pfad an und wählen Sie dann Bibliothek auf S3 hochladen.

   Wenn Sie die Bibliothek bereits hochgeladen haben, wählen Sie Ich habe bereits Bibliotheken hochgeladen und verwenden Sie meinen vorhandenen S3-Bucket auf der Seite zum Hochladen der Bibliothek. Geben Sie als Nächstes den Amazon S3-Pfad an.

   Klicken Sie anschließend auf Next.

5. Wählen Sie auf der Seite Funktionsemulation die Option Erweiterungspaket erstellen aus. Es werden Nachrichten zum Status der Operationen des Erweiterungspakets angezeigt.

   Wenn Sie fertig sind, wählen Sie Finish aus.

Um das Erweiterungspaket beim Anwenden des konvertierten Codes anzuwenden

1. Geben Sie den Amazon S3 S3-Bucket in Ihrem AWS Serviceprofil an. Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.

   Stellen Sie sicher, dass Ihre Amazon S3 S3-Bucket-Richtlinie die folgenden Berechtigungen umfasst:

```
{
    "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"]
      }
    ]
}
```

Ersetzen Sie es im vorherigen Beispiel *111122223333:user/DataExtractionAgentName* durch den Namen Ihres IAM-Benutzers.

2. Konvertieren Sie Ihre Data Warehouse-Quellschemas. Weitere Informationen finden Sie unter Data Warehouse-Schemas konvertieren.
3. Wählen Sie im rechten Bereich das konvertierte Schema aus.
4. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schema-Element und wählen Sie Apply to database aus.
5. AWS SCT generiert Erweiterungspakete mit den erforderlichen Komponenten und fügt das `aws_`*`database_engine_name`*`_ext` Schema in die Zielstruktur ein. AWS SCT Wendet als Nächstes den konvertierten Code und das Erweiterungspaket-Schema auf Ihr Ziel-Data Warehouse an.

   Wenn Sie eine Kombination aus Amazon Redshift und AWS Glue als Zieldatenbankplattform verwenden, AWS SCT fügt dem Erweiterungspaket ein zusätzliches Schema hinzu.

Verwenden der AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket

AWS SCT bietet ein Erweiterungspaket, das Lambda-Funktionen für E-Mail, Jobplanung und andere Funktionen für auf Amazon EC2 gehostete Datenbanken enthält.

Verwendung von AWS Lambda Funktionen zur Emulierung von Datenbankfunktionen

In einigen Fällen können Datenbankfunktionen nicht in äquivalente Amazon RDS-Funktionen konvertiert werden. Beispielsweise sendet Oracle E-Mail-Aufrufe mit `UTL_SMTP`, während Microsoft SQL Server möglicherweise einen Aufgabenplaner verwendet. Wenn Sie eine Datenbank bei Amazon hosten und selbst verwalten EC2, können Sie diese Funktionen emulieren, indem Sie sie durch Dienste ersetzen AWS .

Der AWS SCT Erweiterungspaket-Assistent hilft Ihnen bei der Installation, Erstellung und Konfiguration von Lambda-Funktionen, um E-Mail, Jobplanung und andere Funktionen zu emulieren.

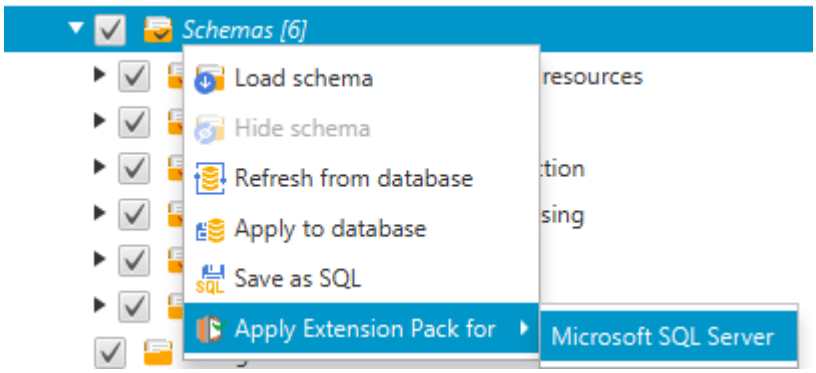
Anwenden des Erweiterungspakets zur Unterstützung von Lambda-Funktionen

Sie können das Erweiterungspaket zur Unterstützung von Lambda-Funktionen mithilfe des Erweiterungspaket-Assistenten anwenden oder wenn Sie den konvertierten Code auf Ihre Zieldatenbank anwenden.

> **Important**
>
> Die Funktionen zur AWS Service-Emulation werden nur für Datenbanken unterstützt, die bei Amazon installiert und selbst verwaltet werden. EC2 Installieren Sie die Service-Emulationsfunktionen nicht, wenn sich Ihre Zieldatenbank auf einer Amazon RDS-DB-Instance befindet.

Um das Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anzuwenden

1. Öffnen Sie in der Zieldatenbankstruktur das Kontextmenü (Rechtsklick), wählen Sie Erweiterungspaket anwenden für und wählen Sie dann Ihre Quelldatenbankplattform aus. AWS Schema Conversion Tool



Der Assistent des Erweiterungspakets wird angezeigt.

2. Lesen Sie die Seite Welcome und wählen Sie dann Next aus.
3. Gehen Sie auf der Seite mit den AWS Profileinstellungen wie folgt vor:

   - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.
   - Wenn Sie AWS Dienste installieren, geben Sie die Anmeldeinformationen ein, um eine Verbindung zu Ihrem herzustellen AWS-Konto. Sie können Ihre AWS CLI Anmeldeinformationen verwenden, wenn Sie die AWS CLI installiert haben. Sie können auch die Anmeldeinformationen verwenden, die Sie zuvor in einem Profil in den globalen Anwendungseinstellungen gespeichert und mit dem Projekt verknüpft haben. Wählen Sie bei Bedarf Navigate to Project Settings aus, um ein anderes Profil mit dem Projekt zu verknüpfen. Wählen Sie bei Bedarf Global Settings aus, um ein neues Profil zu erstellen. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.
4. Führen Sie auf der Seite Email Sending Service folgende Schritte aus:

   - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.
   - Wenn Sie AWS Dienste installieren und bereits über eine Lambda-Funktion verfügen, können Sie diese bereitstellen. Andernfalls erstellt der Assistent eine für Sie. Klicken Sie anschließend auf Next.
5. Führen Sie auf der Seite Job Emulation Service folgende Schritte aus:

   - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.

- Wenn Sie AWS Dienste installieren und bereits über eine Lambda-Funktion verfügen, können Sie diese bereitstellen. Andernfalls erstellt der Assistent eine für Sie. Klicken Sie anschließend auf Next.

6. Wählen Sie auf der Seite Funktionsemulation die Option Erweiterungspaket erstellen aus. Es werden Nachrichten zum Status der Operationen des Erweiterungspakets angezeigt.

   Wenn Sie fertig sind, wählen Sie Finish aus.

> **Note**
>
> Um ein Erweiterungspaket zu aktualisieren und die alten Erweiterungspaket-Komponenten zu überschreiben, stellen Sie sicher, dass Sie die neueste Version von verwenden. AWS SCT Weitere Informationen finden Sie unter Installation und Konfiguration AWS Schema Conversion Tool.

Funktionen für das AWS SCT Erweiterungspaket konfigurieren

Das Erweiterungspaket enthält Funktionen, die Sie vor der Verwendung konfigurieren müssen. Die Konstante `CONVERSION_LANG` definiert die Sprache, die das Service Pack verwendet. Die Funktionen sind für Englisch und Deutsch verfügbar.

Um die Sprache auf Englisch oder Deutsch einzustellen, nehmen Sie die folgende Änderung im Funktionscode vor. Suchen Sie die folgende Konstantendeklaration:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

Um die Einstellung `CONVERSION_LANG` auf Englisch zu ändern, ändern Sie die Zeile wie folgt:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Um die Einstellung `CONVERSION_LANG` auf Englisch einzustellen, ändern Sie die Zeile wie folgt:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Stellen Sie diese Einstellung für die folgenden Funktionen ein:

- 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

Bewährte Methoden für die Verwendung AWS Schema Conversion Tool

Hier finden Sie Informationen zu bewährten Methoden und Optionen für die Verwendung von AWS Schema Conversion Tool (AWS SCT).

Konfiguration von zusätzlichem Speicher

Um große Datenbankschemas zu konvertieren, z. B. eine Datenbank mit 3.500 gespeicherten Prozeduren, können Sie die Speichermenge konfigurieren, die für die zur Verfügung steht. AWS Schema Conversion Tool

Um die Menge an Arbeitsspeicher zu ändern, die verbraucht wird AWS SCT

1. Wählen Sie im Menü Einstellungen die Option Allgemeine Einstellungen und anschließend JVM-Optionen aus.
2. Wählen Sie „Konfigurationsdatei bearbeiten“ und wählen Sie den Texteditor, um die Konfigurationsdatei zu öffnen.
3. Bearbeiten Sie den `JavaOptions`-Abschnitt, um den minimalen und maximalen verfügbaren Speicher festzulegen. Im folgenden Beispiel wird das Minimum auf vier GB und das Maximum auf 40 GB festgelegt.

   ```
   [JavaOptions]
   -Xmx40960M
   -Xms4096M
   ```

   Es wird empfohlen, den verfügbaren Mindestspeicher auf mindestens vier GB festzulegen.
4. Speichern Sie die Konfigurationsdatei, wählen Sie OK und starten Sie neu, AWS SCT um die Änderungen zu übernehmen.

Konfiguration des Standard-Projektordners

AWS SCT verwendet den Projektordner zum Speichern der Projektdateien, zum Speichern von Bewertungsberichten und zum Speichern des konvertierten Codes. AWS SCT Speichert

standardmäßig alle Dateien im Anwendungsordner. Sie können einen anderen Ordner als Standard-Projektordner angeben.

Um den Standard-Projektordner zu ändern

1. Wählen Sie im Menü Einstellungen die Option Allgemeine Einstellungen und dann Dateipfad aus.
2. Geben Sie unter Standardpfad für Projektdateien den Pfad zum Standardprojektordner ein.
3. Wählen Sie Übernehmen und anschließend OK aus.

Erhöhung der Geschwindigkeit der Datenmigration

Um große Datensätze zu migrieren, z. B. eine Gruppe von Tabellen mit mehr als 1 TB Daten, möchten Sie möglicherweise die Migrationsgeschwindigkeit erhöhen. Wenn Sie Datenextraktionsagenten verwenden, hängt die Geschwindigkeit der Datenmigrationen von verschiedenen Faktoren ab. Zu diesen Faktoren gehören die Anzahl der Bereiche in Ihrem Amazon Redshift Redshift-Zielcluster, die Größe einer Chunk-Datei in Ihrer Migrationsaufgabe, der verfügbare Arbeitsspeicher auf dem PC, auf dem Sie Ihre Datenextraktionsagenten ausführen, usw.

Um die Geschwindigkeit der Datenmigration zu erhöhen, empfehlen wir, mehrere Testmigrationssitzungen mit kleinen Datensätzen Ihrer Produktionsdaten durchzuführen. Außerdem empfehlen wir, dass Sie Ihre Datenextraktionsagenten auf einem PC mit einer SSD mit einer Größe von mindestens 500 GB ausführen. Ändern Sie während dieser Testsitzungen verschiedene Migrationsparameter, überwachen Sie Ihre Festplattenauslastung, um herauszufinden, welche Konfiguration die maximale Datenmigrationsgeschwindigkeit gewährleistet. Verwenden Sie dann diese Konfiguration, um Ihren gesamten Datensatz zu migrieren.

Zunehmende Protokollierungsinformationen

Sie können die Anzahl der Protokollierungsinformationen erhöhen, die AWS SCT bei der Konvertierung Ihrer Datenbanken, Skripts und Anwendungs-SQL erzeugt werden. Eine Erhöhung der Protokollierungsinformationen kann zwar die Konvertierung verlangsamen, aber die Änderungen können Ihnen dabei helfen, dem AWS Support zuverlässige Informationen bereitzustellen, falls Fehler auftreten.

AWS SCT speichert Protokolle in Ihrer lokalen Umgebung. Sie können diese Protokolldateien einsehen und sie zur Fehlerbehebung an den AWS Support oder an AWS SCT Entwickler weitergeben.

Um die Protokollierungseinstellungen zu ändern

1. Wählen Sie im Menü Einstellungen die Option Allgemeine Einstellungen und anschließend Protokollierung aus.
2. Geben Sie unter Protokollordnerpfad den Ordner ein, in dem die Protokolle von der Benutzeroberfläche aus gespeichert werden sollen.
3. Geben Sie als Pfad zum Konsolenprotokollordner den Ordner ein, in dem die Protokolle der AWS SCT Befehlszeilenschnittstelle (CLI) gespeichert werden sollen.
4. Geben Sie für Maximale Protokolldateigröße (MB) die Größe einer einzelnen Protokolldatei in MB ein. Wenn Ihre Datei dieses Limit erreicht hat, AWS SCT wird eine neue Protokolldatei erstellt.
5. Geben Sie für Maximale Anzahl von Protokolldateien die Anzahl der zu speichernden Protokolldateien ein. Wenn die Anzahl der Protokolldateien im Ordner diesen Grenzwert erreicht hat, AWS SCT wird die älteste Protokolldatei gelöscht.
6. Geben Sie im Feld Pfad zum Herunterladen des Extraktoren-Protokolls den Ordner ein, in dem die Protokolle der Datenextraktionsagenten gespeichert werden sollen.
7. Geben Sie als Protokollpfad für den Cassandra-Extraktor den Ordner ein, in dem die Protokolle der Apache Cassandra-Datenextraktionsagenten gespeichert werden sollen.
8. Wählen Sie Vor dem Laden nach einem Pfad fragen, um sicherzustellen, dass Sie bei jeder Verwendung von Datenextraktionsagenten AWS SCT gefragt werden, wo die Protokolle gespeichert werden sollen.
9. Wählen Sie für den Debug-Modus True aus. Verwenden Sie diese Option, um zusätzliche Informationen zu protokollieren, wenn die AWS SCT Standardprotokolle keine Probleme enthalten.
10. Wählen Sie die wichtigsten Anwendungsmodule aus, um die Anzahl der Protokollierungsinformationen zu erhöhen. Sie können die Protokollierungsinformationen für die folgenden Anwendungsmodule erhöhen:

    - Allgemeines
    - Loader
    - Parser
    - Drucker
    - Resolver
    - Telemetrie

- Konverter
- Geben Sie die Zuordnung ein
- Benutzeroberfläche
- Steuerung
- Schema vergleichen
- Rechenzentrum klonen
- Anwendungsanalysator

Wählen Sie für jedes der vorherigen Anwendungsmodule eine der folgenden Protokollierungsebenen aus:

- Trace — Die detailliertesten Informationen.
- Debug — Detaillierte Informationen zum Ablauf durch das System.
- Info — Laufzeitereignisse wie Starten oder Herunterfahren.
- Warnung — Verwendung veralteter APIs, schlechter API-Einsatz, andere Laufzeitsituationen, die unerwünscht oder unerwartet sind.
- Fehler — Laufzeitfehler oder unerwartete Bedingungen.
- Kritisch — Fehler, die zum Herunterfahren der Anwendung führen.
- Obligatorisch — Die höchstmögliche Fehlerquote.

Standardmäßig wird nach dem Einschalten des Debug-Modus die Info-Protokollierungsebene für alle Anwendungsmodule AWS SCT festgelegt.

Um beispielsweise bei wichtigen Problemen bei der Konvertierung zu helfen, setzen Sie Parser, Typenzuordnung und Benutzeroberfläche auf Trace.

Wenn die Informationen für das Dateisystem, in dem die Protokolle gestreamt werden, zu umfangreich werden, wechseln Sie zu einem Speicherort mit ausreichend Speicherplatz für die Erfassung von Protokollen.

Um Protokolle an den AWS Support zu übertragen, wechseln Sie in das Verzeichnis, in dem die Protokolle gespeichert sind, und komprimieren Sie alle Dateien in einer einzigen, verwaltbaren ZIP-Datei. Laden Sie dann die ZIP-Datei mit Ihrem Support-Fall hoch. Wenn die erste Analyse abgeschlossen ist und die laufende Entwicklung wieder aufgenommen wird, setzen Sie den Debug-

Modus auf False zurück, um die ausführliche Protokollierung zu vermeiden. Erhöhen Sie dann die Konvertierungsgeschwindigkeit.

> **Tip**
>
> Um die Protokollgröße zu verwalten und die Berichterstattung über Probleme zu vereinfachen, entfernen Sie die Protokolle oder verschieben Sie sie nach einer erfolgreichen Konvertierung an einen anderen Ort. Durch diese Aufgabe wird sichergestellt, dass nur die relevanten Fehler und Informationen an den AWS Support übertragen werden und dass das Protokolldateisystem nicht gefüllt wird.

Behebung von Problemen mit AWS Schema Conversion Tool

Im Folgenden finden Sie Informationen zur Behebung von Problemen mit dem AWS Schema Conversion Tool (AWS SCT).

Objekte können nicht aus einer Oracle-Quelldatenbank geladen werden.

Wenn Sie versuchen, ein Schema aus einer Oracle-Datenbank zu laden, tritt möglicherweise einer der folgenden Fehler auf.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Diese Fehler treten auf, weil der Benutzer, dessen ID Sie für die Verbindung mit der Oracle-Datenbank verwendet haben, nicht über ausreichende Berechtigungen verfügt, um das Schema zu lesen, wie es von erforderlich ist AWS SCT.

Sie können das Problem lösen, indem Sie dem Benutzer die `select_catalog_role`-Berechtigung gewähren und außerdem die Berechtigung für jedes Wörterbuch in der Datenbank. Diese Berechtigungen bieten den schreibgeschützten Zugriff auf die Ansichten und Systemtabellen, der für AWS SCT erforderlich ist. Im folgenden Beispiel wird eine Benutzer-ID mit dem Namen `min_privs` erstellt und dem Benutzer mit dieser ID werden die Minimalberechtigungen gewährt, die für die Konvertierung eines Schemas von einer Oracle-Quelldatenbank erforderlich sind.

```
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;
```

Warnmeldung im Bewertungsbericht

Um die Komplexität der Konvertierung in eine andere Datenbank-Engine beurteilen zu können, AWS SCT ist Zugriff auf Objekte in Ihrer Quelldatenbank erforderlich. Wenn AWS SCT beim Scannen

Probleme auftreten und eine Bewertung nicht durchgeführt werden kann, wird eine Warnmeldung ausgegeben. Diese Meldung weist darauf hin, dass der Prozentsatz der Konversionen insgesamt reduziert wurde. Im Folgenden sind Gründe aufgeführt, aus denen beim Scannen Probleme AWS SCT auftreten können:

- Ihr Datenbankbenutzer hat nicht Zugriff auf alle benötigten Objekte. Weitere Informationen zu den AWS SCT erforderlichen Sicherheitsberechtigungen und Rechten für Ihre Datenbank finden Sie im entsprechenden Abschnitt Herstellen einer Verbindung zu Quelldatenbanken mit dem AWS Schema Conversion Tool zur Quelldatenbank in diesem Handbuch.
- Ein im Schema zitiertes Objekt ist in der Datenbank nicht mehr vorhanden. Um das Problem zu lösen, können Sie mit SYSDBA-Berechtigungen eine Verbindung herstellen und überprüfen, ob das Objekt in der Datenbank vorhanden ist.
- SCT versucht, ein verschlüsseltes Objekt zu bewerten.

CLI-Referenz für AWS Schema Conversion Tool

In diesem Abschnitt werden die ersten Schritte mit der AWS SCT Befehlszeilenschnittstelle (CLI) beschrieben. Außerdem enthält dieser Abschnitt Informationen zu den Tastenbefehlen und Verwendungsmodi. Eine vollständige Referenz der AWS SCT CLI-Befehle finden Sie unterReferenzmaterial.

Themen

- Voraussetzungen für die Verwendung der AWS SCT Befehlszeilenschnittstelle
- AWS SCT Interaktiver CLI-Modus
- AWS SCT CLI-Szenarien abrufen
- AWS SCT CLI-Szenarien bearbeiten
- AWS SCT CLI-Skriptmodus
- AWS SCT CLI-Referenzmaterial

Voraussetzungen für die Verwendung der AWS SCT Befehlszeilenschnittstelle

Laden Sie die neueste Version von Amazon Corretto 11 herunter und installieren Sie sie. Weitere Informationen finden Sie unter Downloads für Amazon Corretto 11 im Amazon Corretto 11-Benutzerhandbuch.

Laden Sie die neueste Version von herunter und installieren Sie sie. AWS SCT Weitere Informationen finden Sie unter Wird installiert AWS Schema Conversion Tool.

AWS SCT Interaktiver CLI-Modus

Sie können die AWS SCT Befehlszeilenschnittstelle im interaktiven Modus verwenden. In diesem Modus geben Sie Befehle nacheinander in die Konsole ein. Sie können diesen interaktiven Modus verwenden, um mehr über CLI-Befehle zu erfahren oder die am häufigsten verwendeten CLI-Szenarien herunterzuladen.

Um Ihr Quelldatenbankschema zu konvertieren AWS SCT, führen Sie einen Sequenzvorgang aus: Erstellen Sie ein neues Projekt, stellen Sie eine Verbindung zu Quell- und Zieldatenbanken

her, erstellen Sie Zuordnungsregeln und konvertieren Sie Datenbankobjekte. Da dieser Workflow komplex sein kann, empfehlen wir die Verwendung von Skripten im AWS SCT CLI-Modus. Weitere Informationen finden Sie unter Skriptmodus.

Sie können die AWS SCT CLI-Befehle im app Ordner Ihres AWS SCT Installationspfads ausführen. In Windows lautet der StandardinstallationspfadC:\Program Files\AWS Schema Conversion Tool\. Stellen Sie sicher, dass dieser Ordner die AWSSchemaConversionToolBatch.jar Datei enthält.

Verwenden Sie den folgenden Befehl, um in den interaktiven AWS SCT CLI-Modus zu wechseln, nachdem Sie die Voraussetzungen erfüllt haben.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Jetzt können Sie AWS SCT CLI-Befehle ausführen. Stellen Sie sicher, dass Sie Ihre Befehle mit / in einer neuen Zeile beenden. Stellen Sie außerdem sicher, dass Sie vor und nach den Werten der Befehlsparameter gerade einfache Anführungszeichen (') verwenden.

> **Note**
>
> Wenn der vorherige Befehl zurückkehrtUnexpected error, versuchen Sie Folgendes:
>
> ```
> java -Djdk.jar.maxSignatureFileSize=20000000 -jar
>  AWSSchemaConversionToolBatch.jar
> ```

Führen Sie den folgenden Befehl aus, um die Liste der verfügbaren Befehle im interaktiven AWS SCT CLI-Modus anzuzeigen.

```
help
/
```

Verwenden Sie den folgenden Befehl, um Informationen zu einem AWS SCT CLI-Befehl anzuzeigen.

```
help -command: 'command_name'
/
```

Ersetzen Sie es im vorherigen Beispiel *command_name* durch den Namen eines Befehls.

Verwenden Sie den folgenden Befehl, um Informationen zu den Parametern eines AWS SCT CLI-Befehls anzuzeigen.

```
help -command: 'command_name' -parameters: 'parameters_list'
/
```

Ersetzen Sie es im vorherigen Beispiel *command_name* durch den Namen eines Befehls. Ersetzen Sie es dann *parameters_list* durch eine Liste von Parameternamen, die durch ein Komma getrennt sind.

Verwenden Sie den folgenden Befehl, um ein Skript aus einer Datei im interaktiven AWS SCT CLI-Modus auszuführen.

```
ExecuteFile -file: 'file_path'
/
```

Ersetzen Sie es im vorherigen Beispiel durch *file_path* den Pfad zu Ihrer Datei mit einem Skript. Stellen Sie sicher, dass Ihre Datei eine `.scts` Erweiterung hat.

Führen Sie den `quit` Befehl aus, um den interaktiven AWS SCT CLI-Modus zu beenden.

Beispiele

Im folgenden Beispiel werden die Informationen über den `Convert` Befehl angezeigt.

```
help -command: 'Convert'
/
```

Im folgenden Beispiel werden die Informationen zu zwei Parametern des `Convert` Befehls angezeigt.

```
help -command: 'Convert' -parameters: 'filter, treePath'
/
```

AWS SCT CLI-Szenarien abrufen

Um die am häufigsten verwendeten AWS SCT Szenarien abzurufen, können Sie den `GetCliScenario` Befehl verwenden. Sie können diesen Befehl im interaktiven Modus ausführen

und dann die heruntergeladenen Vorlagen bearbeiten. Verwenden Sie die bearbeiteten Dateien im Skriptmodus.

Der `GetCliScenario` Befehl speichert die ausgewählte Vorlage oder alle verfügbaren Vorlagen im angegebenen Verzeichnis. Die Vorlage enthält den vollständigen Befehlssatz zum Ausführen eines Skripts. Stellen Sie sicher, dass Sie die Dateipfade, Datenbankanmeldeinformationen, Objektnamen und andere Daten in diesen Vorlagen bearbeiten. Stellen Sie außerdem sicher, dass Sie die Befehle entfernen, die Sie nicht verwenden, und fügen Sie dem Skript bei Bedarf neue Befehle hinzu.

Um den `GetCliScenario` Befehl auszuführen, erfüllen Sie die Voraussetzungen und wechseln Sie in den interaktiven AWS SCT CLI-Modus. Weitere Informationen finden Sie unter interaktiver Modus.

Verwenden Sie als Nächstes die folgende Syntax, um den `GetCliScenario` Befehl auszuführen und die AWS SCT Szenarien abzurufen.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

Ersetzen Sie es im vorherigen Beispiel *`template_type`* durch einen der Vorlagentypen aus der folgenden Tabelle. *`file_path`*Ersetzen Sie als Nächstes den Ordner, in den Sie Skripts herunterladen möchten, durch den Pfad. Stellen Sie sicher, dass AWS SCT Sie auf diesen Ordner zugreifen können, ohne Administratorrechte anzufordern. Stellen Sie außerdem sicher, dass Sie vor und nach den Werten der Befehlsparameter gerade einfache Anführungszeichen (`'`) verwenden.

Um alle AWS SCT CLI-Vorlagen herunterzuladen, führen Sie den vorherigen Befehl ohne die `-type` Option aus.

Die folgende Tabelle enthält die Arten von AWS SCT CLI-Vorlagen, die Sie herunterladen können. Für jede Vorlage enthält die Tabelle den Dateinamen und die Beschreibung der Operationen, die Sie mit dem Skript ausführen können.

Typ der Vorlage	Dateiname	Beschreibung
BTEQScriptKonvertierung	`BTEQScriptConversionTemplate.scts`	Konvertiert Teradata Basic Teradata Query (BTEQ), FastExport, und MultiLoad Skripten in Amazon Redshift FastLoad RSQL. Weitere Informationen finden Sie

Typ der Vorlage	Dateiname	Beschreibung
		unter Daten mithilfe von ETL konvertieren.
ConversionApply	`ConversionTemplate.scts`	Konvertiert Quelldatenbankschemas und wendet den konvertierten Code auf die Zieldatenbank an. Speichert optional den konvertierten Code als SQL-Skript und speichert den Bewertungsbericht. Weitere Informationen finden Sie unter Schemas werden konvertiert.
GenericAppConversion	`GenericApplicationConversionTemplate.scts`	Konvertiert mit dem generischen AWS SCT Anwendungskonverter in Ihre Anwendungen eingebetteten SQL-Code. Weitere Informationen finden Sie unter SQL-Code.
HadoopMigration	`HadoopMigrationTemplate.scts`	Migriert Ihren lokalen Hadoop-Cluster zu Amazon EMR. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Apache Hadoop-Datenbanken mit dem AWS Schema Conversion Tool.

Typ der Vorlage	Dateiname	Beschreibung
HadoopResumeMigration	`HadoopResumeMigrationTemplate.scts`	Setzt eine unterbrochene Migration Ihres lokalen Hadoop-Clusters zu Amazon EMR fort. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Apache Hadoop-Datenbanken mit dem AWS Schema Conversion Tool.
Informatica	`InformaticaConversionTemplate.scts`	Konvertiert SQL-Code, der in Ihre Informatica-Skripts zum Extrahieren, Transformieren und Laden (ETL) eingebettet ist. Konfiguriert Verbindungen zu Ihren Quell- und Zieldatenbanken in Ihren ETL-Skripts und speichert die konvertierten Skripts nach der Konvertierung. Weitere Informationen finden Sie unter Informatica ETL-Skripts.
LanguageSpecificAppConversion	`LanguageSpecificAppConversionTemplate.scts`	Konvertiert mit dem Anwendungskonverter in Ihre C#-, C++-, Java- und Pro*C-Anwendungen eingebetteten SQL-Code. AWS SCT Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren.

Typ der Vorlage	Dateiname	Beschreibung
OozieConversion	`OozieConversionTemplate.scts`	Konvertiert Ihre Apache Oozie-Workflows in. AWS Step Functions Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Apache Oozie-Workflows mit dem AWS Schema Conversion Tool.
RedshiftAgent	`DWHDataMigrationTemplate.scts`	Konvertiert Quell-Data Warehouse-Schemas und wendet den konvertierten Code auf die Amazon Redshift Redshift-Zieldatenbank an. Registriert dann einen Datenextraktionsagenten, erstellt und startet eine Datenmigrationsaufgabe. Weitere Informationen finden Sie unter Migration aus einem Data Warehouse.
ReportCreation	`ReportCreationTemplate.scts`	Erstellt einen Datenbank migrationsbericht für mehrere Quelldatenbankschemas. Speichert diesen Bericht dann als CSV- oder PDF-Datei. Weitere Informationen finden Sie unter Bewertungsbericht.
SQLScriptKonvertierung	`SQLScriptConversionTemplate.scts`	Konvertiert SQL*Plus- oder TSQL-Skripten nach PL/SQL und speichert konvertierte Skripten. Speichert außerdem einen Bewertungsbericht.

Nachdem Sie die AWS SCT CLI-Vorlage heruntergeladen haben, konfigurieren Sie das Skript mit dem Texteditor so, dass es in Ihren Quell- und Zieldatenbanken ausgeführt wird. Verwenden Sie als Nächstes den AWS SCT CLI-Skriptmodus, um Ihr Skript auszuführen. Weitere Informationen finden Sie unter AWS SCT CLI-Skriptmodus.

Beispiele

Im folgenden Beispiel werden alle Vorlagen in den `C:\SCT\Templates` Ordner heruntergeladen.

```
GetCliScenario -directory: 'C:\SCT\Templates'
/
```

Im folgenden Beispiel wird die Vorlage für den `ConversionApply` Vorgang in den `C:\SCT\Templates` Ordner heruntergeladen.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'
/
```

AWS SCT CLI-Szenarien bearbeiten

Nachdem Sie Szenariovorlagen heruntergeladen haben, konfigurieren Sie sie so, dass funktionierende Skripts verfügbar sind, die in Ihren Datenbanken ausgeführt werden können.

Stellen Sie bei allen Vorlagen sicher, dass Sie den Pfad zu den Treibern für Ihre Quell- und Zieldatenbanken angeben. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.

Stellen Sie sicher, dass Sie die Datenbankanmeldedaten für Quell- und Zieldatenbanken angeben. Stellen Sie außerdem sicher, dass Sie Zuordnungsregeln einrichten, um ein Quell-Ziel-Paar für Ihr Konvertierungsprojekt zu beschreiben. Weitere Informationen finden Sie unter Zuordnung von Datentypen.

Als Nächstes konfigurieren Sie den Umfang der auszuführenden Operationen. Sie können die Befehle entfernen, die Sie nicht verwenden, oder dem Skript neue Befehle hinzufügen.

Nehmen wir beispielsweise an, Sie planen, alle Schemas in Ihrer Oracle-Quelldatenbank nach PostgreSQL zu konvertieren. Anschließend planen Sie, Ihren Bewertungsbericht zur Datenbankmigration als PDF zu speichern und den konvertierten Code auf die Zieldatenbank

anzuwenden. In diesem Fall können Sie die Vorlage für den `ConversionApply` Vorgang verwenden. Gehen Sie wie folgt vor, um Ihre AWS SCT CLI-Vorlage zu bearbeiten.

Um die AWS SCT CLI-Vorlage für den **`ConversionApply`** Vorgang zu bearbeiten

1. Öffnen Sie die Datei`ConversionTemplate.scts`, die Sie heruntergeladen haben. Weitere Informationen finden Sie unter Beispiele.
2. Entfernen CreateFilterund konvertieren Sie -filter-, ApplyToTarget -filter -, SaveTargetSQL -, SaveTargetSQLbyStatement - und SaveReportCSV-Operationen aus Ihrem Skript.
3. Geben Sie für oracle_driver_file im SetGlobalSettingsVorgang den Pfad zu Ihrem Oracle-Treiber ein. Geben Sie dann für postgresql_driver_file den Pfad zu Ihrem PostgreSQL-Treiber ein.

   Wenn Sie andere Datenbank-Engines verwenden, verwenden Sie die entsprechenden Namen für die Einstellungen. Eine vollständige Liste der globalen Einstellungen, die Sie während des SetGlobalSettingsVorgangs festlegen können, finden Sie unter Globale Einstellungsmatrix in derReferenzmaterial.
4. (Optional) Geben Sie für CreateProjectden Namen Ihres Projekts und den Speicherort Ihrer lokalen Projektdatei ein. Wenn Sie mit den Standardwerten fortfahren möchten, stellen Sie sicher, dass Dateien in dem `C:\temp` Ordner erstellt werden AWS SCT können, ohne Administratorrechte anzufordern.
5. Geben Sie für AddSourcedie IP-Adresse Ihres Quelldatenbankservers ein. Geben Sie außerdem den Benutzernamen, das Passwort und den Port ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.
6. Geben Sie für AddTargetdie IP-Adresse Ihres Zieldatenbankservers ein. Geben Sie außerdem den Benutzernamen, das Passwort und den Port ein, um eine Verbindung zu Ihrem Zieldatenbankserver herzustellen.
7. (Optional) Geben Sie für AddServerMappingdie Quell- und Zieldatenbankobjekte ein, die Sie einer Zuordnungsregel hinzufügen möchten. Sie können `targetTreePath` Parameter `sourceTreePath` und verwenden, um den Pfad zu den Datenbankobjekten anzugeben. Optional können Sie `sourceNamePath` und verwenden, `targetNamePath` um die Namen der Datenbankobjekte anzugeben. Weitere Informationen finden Sie unter Befehle zur Serverzuordnung in derReferenzmaterial.

   Die Standardwerte des AddServerMappingVorgangs ordnen alle Quellschemas Ihrer Zieldatenbank zu.

8. Speichern Sie die Datei und verwenden Sie dann den Skriptmodus, um sie auszuführen. Weitere Informationen finden Sie unter Skriptmodus.

AWS SCT CLI-Skriptmodus

Nachdem Sie ein AWS SCT CLI-Skript erstellt oder eine Vorlage bearbeitet haben, können Sie es mit dem `RunSCTBatch` Befehl ausführen. Stellen Sie sicher, dass Sie Ihre Datei mit dem CLI-Skript als `.scts` Erweiterung speichern.

Sie können AWS SCT CLI-Skripte aus dem `app` Ordner Ihres AWS SCT Installationspfads ausführen. In Windows lautet der Standardinstallationspfad`C:\Program Files \AWS Schema Conversion Tool\`. Stellen Sie sicher, dass dieser Ordner die `RunSCTBatch.sh` Datei `RunSCTBatch.cmd` oder enthält. Außerdem sollte dieser Ordner die `AWSSchemaConversionToolBatch.jar` Datei enthalten.

Alternativ können Sie den Pfad zur `RunSCTBatch` Datei in der `PATH` Umgebungsvariablen Ihres Betriebssystems hinzufügen. Nachdem Sie die `PATH` Umgebungsvariable aktualisiert haben, können Sie AWS SCT CLI-Skripts von jedem Ordner aus ausführen.

Verwenden Sie den folgenden Befehl in Windows, um ein AWS SCT CLI-Skript auszuführen.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Ersetzen Sie es im vorherigen Beispiel durch *`file_path`* den Pfad zu Ihrer Datei mit einem Skript.

Verwenden Sie den folgenden Befehl unter Linux, um ein AWS SCT CLI-Skript auszuführen.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Ersetzen Sie im vorherigen Beispiel durch *`file_path`* den Pfad zu Ihrer Datei mit einem Skript.

Sie können in diesem Befehl optionale Parameter angeben, z. B. Datenbankanmeldedaten, den Detaillierungsgrad in der Konsolenausgabe und andere. Weitere Informationen finden Sie in der Referenz zur AWS SCT Befehlszeilenschnittstelle unterReferenzmaterial.

Beispiele

Im folgenden Beispiel wird das `ConversionTemplate.scts` Skript im `C:\SCT\Templates` Ordner ausgeführt. Sie können dieses Beispiel in Windows verwenden.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

Im folgenden Beispiel wird das `ConversionTemplate.scts` Skript im `/home/user/SCT/Templates` Verzeichnis ausgeführt. Sie können dieses Beispiel unter Linux verwenden.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT CLI-Referenzmaterial

Referenzmaterial zur AWS Schema Conversion Tool Befehlszeilenschnittstelle (CLI) finden Sie in der folgenden Anleitung: AWS Schema Conversion Tool CLI Reference.

Versionshinweise für AWS Schema Conversion Tool

Dieser Abschnitt enthält Versionshinweise für AWS SCT, beginnend mit Version 1.0.640.

Versionshinweise für AWS SCT Build 676

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Oracle	PostgreSQL/ Aurora Postgre L	Neue integrierte Funktionsemulation für die folgenden Funktionen: • `SYS.UTL_RAW.BIT_AND(RAW,RAW)` • `XDB.DBMS_XSLPROCESSOR.CLOB2FILE(CLOB)` • `XDB.DBMS_XSLPROCESSOR.READ2CLOB(VARCHAR2)` • `SYS.UTL_RAW.BIT_OR(RAW,RAW)` • `SYS.UTL_RAW.BIT_COMPLEMENT(RAW)`	Nein	Ja
MS SQL Server	Amazon RDS-SQL-Server	`Database Mail not supported` Nachricht aus dem PDF-Bericht entfernt	Ja	Ja
Oracle	PostgreSQL/ Aurora	Die Konvertierung von Einschränkungen für partitionierte Tabellen wurde implementiert.	Ja	Ja

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
	Postgre L			
Oracle	MySQL	Überprüfung der Anwendbarkeit von AI-602 bei der Tabellenkonvertierung	Ja	Ja
MS SQL Server	Postgre L/ Aurora Postgre L	unterstützt jetzt das `MERGE` Statement in PostgreSQL 15.x	Ja	Ja
Alle	Alle	Implementierte JDBC-Verbindungen: Erweiterte Eigenschaften	Ja	Nein
Alle	Alle	CLI: `PrintOLAPTaskStatus` Befehlsfehler behoben	Ja	Nein
Teradat	Amazor Redshif	Datentypumwandlung im Teradata-Stil implementiert.	Ja	Nein
Teradat	Amazor Redshif	Fehlerhafte Konvertierung in SQL/BTEQ `MERGE` behoben.	Ja	Nein
Teradat	Amazor Redshif	Datentypumwandlung im Teradata-Stil implementiert.	Ja	Nein
Teradat	Amazor Redshif	Funktionsumwandlungen wurden implementiert. `LEAD/LAG`	Ja	Nein
Teradat	Amazor Redshif	Fehler behoben`AI-9996 Transformer error occurred in statement .`	Ja	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Teradat	Amazor Redshif	Fehler behobenAI-9996 Transformer error in selectItem .	Ja	Nein
Teradat	Amazor Redshif	Konvertierung für eine teilweise gespeicherte Prozedur implementiert: XbiDQM.SpCmprsnDly	Ja	Nein
Teradat	Amazor Redshif	UNPIVOTAnweisung mit Alias implementiert.	Ja	Nein
Teradat	Amazor Redshif	Implementierte Delete Anweisung mit mehreren Quelltabellen.	Ja	Nein
Teradat	Amazor Redshif	Fix fürAI-9996 Transformer error occurred in functionCallExpression .	Ja	Nein
Teradat	Amazor Redshif	Die Konvertierung von NORMALIZE Klauseln wurde implementiert.	Ja	Nein
Teradat	Amazor Redshif	Die falsche Konvertierung in DELETE Anweisungen mit Unterabfragen wurde behoben.	Ja	Nein
Teradat	Amazor Redshif	Fehler AI-9996 Transformer error occurred in tableOperatorSource behoben.	Ja	Nein
Teradat	Amazor Redshif	Fehler behobenAI-9996 Transformer error occurred in additiveExpression .	Ja	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Teradat	Amazor Redshif	Die Konvertierung von DBC-Systemobjekten wurde implementiert.	Ja	Nein
Teradat	Amazor Redshif	Die Problemumgehung des Updates mit impliziten Join-Prädikaten wurde implementiert.	Ja	Nein
Netezza	Amazor Redshif	Fehler bei der Konvertierung von Anweisungen behobenCREATE MATERIALIZED VIEW .	Ja	Nein
Db2LUV	PostgreL/ Aurora PostgreL	Verbindung mit erweiterten JDBC-Optionen: Zusätzliche Verbindungsoptionen hinzugefügt.	Ja	Nein
DB2LUV	PostgreL/ Aurora PostgreL	Unterstützung für MERGE Statement in PostgreSQL 15.x hinzugefügt	Ja	Nein
Db2LUV	PostgreL/ Aurora PostgreL	Konvertierung GLOBAL TEMPORARY TABLE implementiert.	Ja	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Db2LUV	PostgreL/ Aurora PostgreL	Konvertierung `USER DEFINED TYPES` implementiert.	Ja	Nein
Db2LUV	MySQL	Konvertierung implementiert. `GLOBAL TEMPORARY TABLE`	Ja	Nein
Db2LUV	MySQL	Konvertierung implementiert. `USER DEFINED TYPES`	Ja	Nein
Db2LUV	MySQL	Konvertierung implementiert. `USER DEFINED FUNCTIONS`	Ja	Nein
Db2LUV	MariaDE	Konvertierung implementiert. `GLOBAL TEMPORARY TABLE`	Ja	Nein
Db2LUV	MariaDE	Konvertierung implementiert. `USER DEFINED TYPES`	Ja	Nein
Sybase	Alle	Unterstützung für Kerberos-Authentifizierung hinzugefügt	Ja	Nein
DB2LUV	PostgreL/ Aurora PostgreL	Unterstützung für die Konvertierung mehrerer Versionen für die Ziele wurde hinzugefügt	Ja	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit im AWS DMS Schema Conversion Tool (SCT)	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Azure SQL/ Microso SQL Server	Postgre L/ Aurora Postgre L	Unterstützung für die Konvertierung mehrerer Versionen für die Ziele wurde hinzugefügt	Ja	Nein
DB2LU\	Postgre L/ Aurora Postgre L	Unterstützung für `MERGE` Statements in PostgreSQL 15.x hinzugefügt.	Ja	Nein
Teradat	Amazor Redshif	Die nicht unterstützte Konvertierung von Funktionsänderungen wurde behoben.	Ja	Nein
Alle	Amazor Redshif	Datenextraktoren: Partitionierung durch eine indizierte Spalte implementiert.	Ja	Nein

Versionshinweise für Build 675 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Cassandr	DynamoD	Es wurde ein Fehler behoben, bei dem die Installation von Cassandra im Zielrechenzentrum fehlschlug.	Nein
DB2 LUW	PostgreS L	DYNAMIC SQL: PREPARE-Anweisung: Auflösung und Konvertierung ohne dynamisches SQL.	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 LUW	PostgreSQL	Unterstützung für SPECIAL REGISTER hinzugefügt.	Nein
DB2 LUW	PostgreSQL	Aktualisierung des Erweiterungspakets	Nein
Hadoop	Amazon EMR	Unterstützung für die Verbindung zu einem Hadoop-Cluster über das rsa-sha2-Protokoll wurde hinzugefügt.	Nein
Microsoft SQL Server	Amazon Redshift	Korrektur für den JDBC-Treiber, der TLS erzwingt, obwohl er nicht konfiguriert ist.	Nein
Netezza	Amazon Redshift	Unterstützung für die Konvertierung von Materialized Views wurde hinzugefügt.	Nein
Oracle	Amazon Redshift	Unterstützung für rekursive Abfragen in Amazon Redshift hinzugefügt.	Ja
Oracle	PostgreSQL, Aurora PostgreSQL	Die falsche Konvertierung des NUMBER-Datentyps wurde behoben.	Ja
Oracle	Amazon Redshift	Datenmigration. Automatische Partitionierung von Oracle. Die Ablaufzeit für den Wert für Tabellenfragmente wurde hinzugefügt. Die Ablaufzeit beträgt 72 Stunden. Wenn ein Ablauf eintritt, werden Datenfragmente neu erstellt, wenn eine Datenmigrationsaufgabe erstellt wird.	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Oracle	Amazon Redshift	SCT Data Extractor: Der Ansatz zum Hochladen von Daten auf Amazon Redshift wurde geändert. Standardmäßig erstellt der Extraktor keine Staging-Tabellen. Nachdem sich alle Datendateien im Amazon S3 S3-Bucket befinden, kopiert der Extraktor sie stattdessen mit einem einzigen COPY-Befehl in die Zieltabelle.	Nein
Oracle	Amazon Redshift	Migration des RAW-Datentyps zur VARBYTE-Spalte hinzugefügt.	Nein
Oracle	PostgreSQL, Aurora PostgreSQL	Konvertierung mehrerer Versionen	Nein
Oracle	PostgreSQL	Unterstützung für MERGE Statement in PostgreSQL 15.x hinzugefügt.	Ja
Oracle	PostgreSQL	Unterstützung für neue Funktionen für reguläre Ausdrücke in PostgreSQL 15.x hinzugefügt.	Ja
Oracle	PostgreSQL, Aurora PostgreSQL	Die ON CONFLICT DO UPDATE-Anweisung wird ohne ausgeschlossenen Alias konvertiert.	Ja
Teradata	Amazon Redshift	Konvertierungsunterstützung für LEAD/LAG-Funktionen hinzugefügt.	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Teradata	Amazon Redshift	Verbesserte Datentypumwandlung mit expliziter Angabe des Datenformats.	Nein
Teradata	Amazon Redshift	Verbesserte Konvertierung der AT 'TIME ZONE'-Klausel in Zeit-/Zeitstempelausdrücken.	Nein
Teradata	Amazon Redshift	AI-9996 bei Konvertierungsvorgängen mit MERGE-Anweisungen.	Nein

Versionshinweise für Build 674 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Alle	Alle	Verschiedene Fehlerkorrekturen und Leistungsverbesserungen	Teilweise (nur für unterstützte Quell- und Zielpaare)
Azure SQL/ Microsoft SQL Server	Amazon Redshift	Die Meldung „AI 18066: Der Schemaname kann nicht konvertiert werden“, die Benutzer bei der Schemabewertung/-konvertierung irreführte, wurde entfernt	Nein
Azure SQL/ Microsoft SQL Server	Amazon RDS for MySQL// Amazon	Fehlerhafte Konvertierung des Verfahrens ohne Zuweisung eines Returncodes	Teilweise (Die Schemakonvertierung unterstützt Azure SQL

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
	Aurora MySQL		derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	In einigen Fällen der Konvertierung von AI9997 FOR XML PATH-Klauseln wurde ein Problem behoben	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Der Wert wird im Prozedur-/Funktionstext auf die ursprüngliche Skala gerundet	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Verbesserungen bei der Konvertierung von EXECUTE-Anweisungen	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Azure SQL/ Microsoft SQL Server/ Azure Synapse	Amazon Redshift	Die Konvertierung der folgenden Anweisungen und Modi wurde verbessert: • AUSNAHMEBLOCK • AUTOCOMMIT • NONATOMIC • GRUPPIERUNGSSATZ • CUBE • ROLLUP	Nein
DB2 LUW	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Korrekturen beim Laden von SQL-Abfragen für Metadaten	Nein
DB2 LUW	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	AI 9996 wird bei Triggern nicht erwartet	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Analytische Funktion ROWNUMBER	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Unterstützung für hexadezimale Zeichenkettenkonstanten	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Korrekturen beim Laden von SQL-Abfragen für Metadaten.	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	NÄCHSTER WERT FÜR die Unterstützung von Sequenzreferenzen	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Unterstützung der _NUMBER_ROWS-Option der GET DIAGNOSTICS-Anweisung DB2	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	GET DIAGNOSTICS Mehrere Anweisungen	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Fehler bei der Konvertierung von FOR-Anweisungen wurden behoben.	Nein
Oracle	Amazon RDS for MySQL// Amazon Aurora MySQL	Ein Fehler wurde behoben, bei dem der Parameterknoten der Paketfunktion nicht definiert war.	Ja
Oracle	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Fehler in der Funktion _EXT.NEXT_DAY des Erweiterungspakets wurden behoben AWS_ORACLE	Ja

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Oracle	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Fehler bei der Konvertierung „(+)“ in den Outer Joins von Oracle wurden behoben	Ja
Oracle		Support der Kerberos-Authentifizierung	Nein
SAP ASE	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Ein Fehler beim Konvertieren von mehr als einem Bezeichner in der FROM-Klausel in der UPDATE-Anweisung wurde behoben	Nein
SAP ASE	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Ein Fehler bei der Konvertierung mehrzeiliger Kommentare und Anweisungen wurde behoben	Nein
SAP ASE		Unterstützung für den Parameter ENCRYPT_PASSWORD beim Herstellen einer Verbindung hinzugefügt	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Teradata	Amazon Redshift	Die Konvertierung der VOLATILE-Tabelle mit dem angegebenen Schemanamen wurde verbessert	Nein
Teradata	Amazon Redshift	Falsche Konvertierung WHERE-KLAUSEL in komplexem CTE	Nein
Teradata	Amazon Redshift	Unterstützung für den INTERVAL-Datentyp bei der Migration von Daten mit SCT-Datenextraktionsagenten wurde hinzugefügt.	Nein
Teradata BTEQ-Skripte	Amazon Redshift RSQL-Skripts	Falsche Ausgangsparameter für die Konvertierung in der von BTEQ ausgeführten Prozedur	Nein

Versionshinweise für Build 673 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Allgemeine Bugfixes und Leistungsverbesserungen
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Fehlerhafte Konvertierung von Funktionsaufrufen wurde behoben
Azure SQL/ Microsoft	Aurora PostgreSQL/ Amazon RDS	Die Konvertierung der XML-Klausel wurde implementiert `F0R`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	PostgreSQL	
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Konvertierung der `FOR` XML-Klausel mit falschem Alias.
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, der AWS SCT auftrat, wenn `EXECUTE` Anweisungen, die eine Zeichenfolge mit Prozedurparametern ausführen, nicht konvertiert wurden.
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Verbesserte Konvertierung von UPDATE-Anweisungen mit inneren Joins.
Azure Synapse	Amazon Redshift	Die falsche Konvertierung der `OBJECT_ID` integrierten Funktion wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM DB2 für z/OS	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Die Konvertierung der folgenden Anweisungen und Objekte wurde implementiert: • `DECLARE TEMPORARY TABLE statement` • `DROP TABLE statement` • PKund `UNIQUE` Einschränkungen für partitionierte Tabellen • `TIMESTAMPDIFF` Funktion • `TO_DATE` Funktion • `EBCDIC_STR` Funktion • `VARCHAR_FORMAT` Funktion
IBM DB2 für z/OS	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, bei dem der funktionsbasierte Index die Funktionen nach der Konvertierung übersprungen hat.
IBM DB2 für z/OS	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, bei dem die `REPEAT` Anweisung nach der Konvertierung mit AI 9996 geschlossen wurde
IBM DB2 für z/OS	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Ein Fehler wurde behoben, bei dem die Klausel mit 9996 geschloss en wurde. `FINAL TABLE`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM DB2 für z/OS	Aurora PostgreSQL/ Amazon RDS PostgreSQL	LOADER| Einschränkung für die Partitionierung des Schlüssels in den Referenzen. AWS SCT kann jetzt Primärschlüssel und eindeutige Einschränkungen in partitionierten Tabellen als Sekundärindizes konvertieren.
IBM für z/OS DB2	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Unterstützung der Funktion Postgresql.varchar_format
IBM für z/OS DB2	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Die Kollationsänderung `CreateTransformationRule` und die `ModifyTransformationRule` SCT-CLI-Befehle wurden implementiert.
Greenplum	Amazon Redshift	Der Fehler mit dem falschen Aufruf der gespeicherten Prozedur nach der Konvertierung wurde behoben
Hadoop	Amazon EMR	Unterstützung für die Verbindung zu einem Hadoop-Cluster über das rsa-sha2-Protokoll hinzugefügt.
Hadoop	Amazon EMR	Unterstützung für Amazon EMR mit Metastore ohne Glue Hive hinzugefügt,
Oracle	Amazon Redshift	Ein Fehler bei der falschen Konvertierung einer rekursiven Abfrage, bei der die Spalte PRIOR nicht in der SELECT-Liste enthalten war, wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Implementiert, die ein Element eines assoziativen Arrays zurückgibt
Oracle	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Unerwartetes AI 9996 in Klammern behoben `UNPIVOT`
Oracle	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Unerwartetes AI 9996 in mit behoben `UNPIVOT` `UNION ALL`
Oracle	Aurora PostgreSQL/ Amazon RDS PostgreSQL	Verbesserungen für Datentypkonvertierungen `Number`
Oracle	Amazon Redshift Datenextraktor	Support für die automatische Partitionierung von Oracle-Tabellen. Optimierung für die Erstellung von Migrationsaufgaben.
Teradata	Amazon Redshift	Implementieren Sie die Konvertierung der `EXCEPTION BLOCK` Aussage

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Support für die Konvertierung von `ALLANY`, und `SOME` Prädikaten nach Amazon Redshift.
Teradata	Amazon Redshift	Native Unterstützung für Prädikate hinzugefügt. `QUALIFY`
Teradata	Amazon Redshift	Die Konvertierung der folgenden Elemente wurde verbessert: • Rekursive Abfragen • `GROUPING SET` • `CUBE` • `ROLLUP` • `UPDATE`Anweisung mit impliziter Verknüpfung
OLAP-Quellen	Amazon Redshift Datenextraktor	CLI-Befehle für Stop/Resume für Amazon Redshift Data Extractor-Aufgaben implementiert.
OLAP-Quellen	Amazon Redshift Datenextraktor	Es wurde die Möglichkeit hinzugefügt, die Tabellenspalten auszuwählen, die während der Konfiguration der Migrationsaufgabe migriert werden müssen.

Versionshinweise für AWS SCT Build 672

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon RDS für PostgreSQL	Unterstützung der PostgreSQL-Hauptversion 15 als Migrationsziel implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Es wurde ein neuer `PrintTaskStatus` Befehl in der AWS SCT Befehlszeilenschnittstelle (CLI) hinzugefügt, um den Status der Datenmigrationsaufgabe anzuzeigen.
Alle	Amazon Redshift	Der Konfigurationsablauf für die Datenextraktionsagenten wurde verbessert.
Alle	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Datenextraktionsagenten die Informationen zu Unteraufgaben nicht anzeigten.
Apache Oozie	AWS Step Functions	Es wurde eine Option hinzugefügt, um State-Machine-Definitionen als Skript im konvertierten Code zu speichern.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von`COALESCE`, `DATEADDGETDATE`, und `SUM` Funktionen wurde implementiert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `UPDATE` Aussagen mit `JOIN` und `OUTPUT` -Klauseln wurde verbessert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, der bei der Konvertierung der `SELECT TOP 1 WITH TIES` Anweisung aufgetreten ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurden mehrere Probleme behoben, die bei der Konvertierung der `FOR XML` Klauseln in integrierten Funktionen auftraten.
Greenplum	Amazon Redshift	Die Konvertierung von `GET DIAGNOSTICS` `RAISE` AND-Anweisungen wurde mithilfe eines nativen Amazon Redshift `EXCEPTION` Redshift-Blocks implementiert.
Greenplum	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren wurde verbessert, indem die Unterstützung eines `EXCEPTION` Blocks im konvertierten Code hinzugefügt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem `TO_CHAR` Funktionen mit Zeitformatvorlagen falsch konvertiert wurden.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung verschachtelter Tabellenausdrücke wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung der Anweisungen`GOTO`, `MERGEREPEAT`, und `SIGNAL` wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `FETCH` Anweisungen mit Schlüsselwörtern `BEFORE` und `AFTER` Orientierungsschlüsselwörtern wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung der `OLD TABLE` Tabellenverweise `FINAL TABLE` und.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung der folgenden Funktionen wurde implementiert. • ADD_MONTHS • DAYmit Parametern des Zeichendatentyps • DAYOFWEEK • DAYS • DECODE • HOUR • LAST_DAY • LOCATE_IN_STRING • MICROSECOND • MINUTE • MONTH • ROUND • TIME • TIMESTAMP • TIMESTAMP_FORMAT • TRANSLATE • UNICODE_STR • XMLCAST • XMLELEMENT • XMLQUERY • XMLSERIALIZE • YEAR
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung eines Alias einer Unterabfrage in Klauseln wurde verbessert. JOIN

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `COALESCE` Funktionen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `EXPLICIT` Indizes wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Spaltennamen in zusammengesetzten Ausdrücken wurde verbessert, um ein Problem zu beheben, bei dem das Aktionselement 9997 während der Konvertierung unerwartet angezeigt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Primärschlüsseln und eindeutigen Einschränkungen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `XMLTABLE` Anweisungen in `INSERT` Anweisungen wurde verbessert, um ein Problem zu beheben, bei dem während der Konvertierung unerwartet der Aktionspunkt 9996 angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Funktionen mit dem Argument angezeigt wurde. `SUBSTR`
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung des `CURRENT TIMESTAMP` Spezialregisters angezeigt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet bei der Konvertierung von `MERGE` Anweisungen, nicht unterstützten Anweisungen und nicht unterstützten integrierten Funktionen angezeigt wurde.
Microsoft SQL Server	Alle	Unterstützung für Microsoft SQL Server Version 2022 als Quelle hinzugefügt.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `SELECT` Anweisungen, die Operatoren zur Verkettung von Zeichenketten verwenden, wurde verbessert. AWS SCT verwendet die `STRING_AGG` Funktion im konvertierten Code.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version 3.1.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsag enten die Datenmigration nicht vom angegebenen CDC-Punkt aus starteten.
Oracle	Alle	Der Bewertungsbericht für Oracle-Datenbanken Version 19 wurde als Quelle aktualisiert.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung des `DBMS_OUTPUT` Pakets wurde implementiert, indem dem AWS SCT Erweiterungspaket neue Funktionen hinzugefü gt wurden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Funktionen und Prozeduren, die assoziative Arrays als Argumente oder Parameter verwenden, wurde implement iert.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `DISTINCT` Klauseln in Anweisungen wurde verbessert. `SELECT`
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Tabellen, bei denen die Primärschlüsselein schränkung denselben Namen wie die Tabelle hat, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung der `RAISE_APPLICATION_ERROR` Prozedur mit dem dritten Parameter wurde verbessert.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem die Migrationsregel den `NUMERIC` Datentyp, `INTEGER` sofern zutreffend, nicht automatisch geändert hat.
Oracle DW	Amazon Redshift	Unterstützung für native Amazon Redshift `CONNECT BY` Redshift-Klauseln im konvertierten Code implementiert.
Oracle DW	Amazon Redshift	Die Datenmigration wurde verbessert, indem automatisch eine Unteraufgabe für jede Tabelle oder Partition im Migrationsbereich hinzugefügt wurde. Dieser Ansatz verhindert den Datenverlust eingefügter Daten nach der Partitionierung.
Teradata	Amazon Redshift	Die Konvertierung von rekursiven Ansichten wurde implementiert.
Teradata	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren, die `ANSI` Transaktionsmodi verwenden`BTET`, wurde verbessert, indem Unterstützung für den nativen Amazon Redshift `AUTOCOMMIT` Redshift-Transaktionsmodus hinzugefügt wurde.
Teradata	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren, die `TERADATA` Transaktionssemantik verwenden, wurde verbessert, indem das `NONATOMIC` Schlüsselwort dem konvertierten Code hinzugefügt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem der konvertierte Code die AWS Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel enthielt.

Versionshinweise für AWS SCT Build 671

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde ein Fehler behoben, bei dem AWS SCT ich nicht berechtigt war, eine Projektdatei in Windows zu speichern.
Alle	Alle	Die folgenden Vorlagen für die AWS SCT Befehlszeilenschnittstelle (CLI) wurden aktualisiert. • BTEQScriptKonvertierung • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatica Weitere Informationen zu den AWS SCT CLI-Vorlagen finden Sie unterCLI-Szenarien abrufen.
Alle	Amazon Redshift	Es wurde ein Fehler behoben AWS SCT , bei dem kein Erweiterungspaket in der Befehlszeilenschnittstelle (CLI) erstellt wurde.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem AWS SCT Datenextraktionsagenten die AWS Snowball Edge Konfiguration in der Befehlszeilenschnittstelle (CLI) nicht verwendeten.
Apache Oozie	AWS Step Functions	Unterstützung für die Migration von Apache Oozie zum AWS Step Functions Befehlszeilenschnittstellen-Modus (CLI) implementiert. Nachdem Sie Ihre Hadoop-Workloads zu Amazon EMR migriert haben, können Sie das Workflow-Scheduling-System nun auf das

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		migrieren. AWS Cloud Weitere Informationen finden Sie unter Konvertierung von Oozie-Workflows;.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Resolver-Fehler behoben, der für Tabellen und Aliase auftrat.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `INDEX ON` Klauseln wurde implementiert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung der folgenden Objekte wurde verbessert, um unerwartete Aktionselemente zu vermeiden. • Batch-Kontoauszüge • Listen von Ausdrücken • Aliasnamen für Tabellen • Temporäre Tabellen • Auslöser • Benutzervariablen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Analysefehler behoben, der bei Prozeduren aufgetreten ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem falsche Namen temporärer Tabellen im konvertierten Code für `OBJECT_ID` Funktionen AWS SCT verwendet wurden.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurden Probleme behoben, bei denen das Aktionselement 9996 während der Konvertierung der folgenden Codeelemente unerwartet angezeigt wurde. • `CONVERT`-Funktionen • `DATEADD`-Funktionen • `DELETE`Anweisungen innerhalb von Inline-Funktionen • `IF`-Anweisungen. • `INSERT`oder `UPDATE` Aktionen in einer Spalte • `RETURN`-Anweisungen. • `UPDATE`Anweisungen mit komplexen Abfragen oder Funktionen
BigQuery	Amazon Redshift	Unterstützung für BigQuery als Quelle für den Multiserver-Assessment-Prozess hinzugefügt. Weitere Informationen finden Sie unter Bewertungsbericht für mehrere Server.
Hadoop	Amazon EMR	Die Version des unterstützten Apache Hive JDBC-Treibers, den Sie für die Verbindung zu Ihren Quelldatenbanken verwenden, wurde aktualisiert. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Quelldatenbankobjekte wie Primärschlüssel, implizite Indizes usw. AWS SCT geladen werden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Resolver-Fehler behoben, der für Spalten in impliziten Cursorn auftrat.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit implementiert, die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen im konvertierten Code beizubehalten.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von schemaübergreifenden Fremdschlüsseln wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `LENGTH` und `VARCHAR` Funktionen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `LABEL ON` AND-Anweisungen wurde implementiert. `DECLARE CONDITION`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `SELECT` Anweisungen mit `OPTIMIZE FOR` Klauseln wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `CREATE TABLE` Anweisungen wurde verbessert, indem Standardwerte für alle unterstützten Datentypen hinzugefügt wurden.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `INCREMENT BY` Attributen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von partitionierten Tabellen wurde verbessert, indem die Möglichkeit hinzugefügt wurde, Tabellenpartitionen vom Konvertierungsbereich auszuschließen.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Primärschlüsseldefinitionen mit `INCLUDE` Spalten wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung der Funktion wurde verbessert. `SUBSTRING`
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `SET` UND-Anweisungen wurde verbessert. `DECLARE HANDLER FOR`
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung variabler Datentypen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `XMLTABLE` Funktionen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Der Migrationsablauf wurde verbessert, indem die folgende Reihenfolge implementiert wurde, in der konvertierte Objekte auf die Zieldatenbank angewendet werden: Tabellen, Partitionen, Indizes, Einschränkungen, Fremdschlüssel und Trigger.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Kommentaren im Quellcode angezeigt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9997 unerwartet bei der Konvertierung von Aliasnamen in Klauseln angezeigt wurde. `FROM`
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9997 unerwartet während der Konvertierung von Cursor-Aliasen angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem der konvertierte Code unterschiedliche Ergebnisse für `SELECT` Anweisungen mit `ORDER BY` Klauseln zurückgab. Da SQL Server und PostgreSQL `NULL` Werte unterschiedlich behandeln, enthält der konvertierte Code jetzt `NULLS FIRST` `NULLS LAST` OR-Klauseln, die sicherstellen, dass Ihr konvertierter Code Ergebnisse in derselben Reihenfolge zurückgibt wie Ihr Quellcode.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Datentypen in Tabellenf unktionen falsch konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
MySQL	Amazon RDS für MySQL	Es wurde ein Problem behoben, bei dem unerwartet einfache Anführungszeichen (`' '`) in den Namen der Datenbankobjekte im konvertierten Code auftauchten.
Oracle	Aurora PostgreSQL PostgreSQL	Dem Erweiterungspaket wurden neue Ansichten hinzugefügt, um Oracle-Systemansichten zu emulieren, mit denen Sie Informationen über Partitionen und Unterpartitionen anzeigen.
Oracle	Aurora PostgreSQL PostgreSQL	Zwei Funktionen im Erweiterungspaket wurden aktualisiert, um Schemanamen als Argumente im konvertierten Code hinzuzufügen.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem nach dem Aktualisieren des Anwendungscodes in der Benutzeroberfläche AWS SCT nicht die richtigen Parameter für die Konvertierung von C++-Anwendungen verwendet wurden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `CREATE TYPE` Anweisungen wurde verbesser t, um unerwartete Ausnahmen zu vermeiden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von verschachtelten Tabellen wurde verbessert.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Analysefehler behoben, der bei Paketobjekten aufgetreten ist.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Objektnamen im konvertierten Code AWS SCT unerwartet gekürzt wurden, wenn die Namenslänge 60 Zeichen überschreitet.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Trigger auf Zeilenebene für partitionierte Tabellen falsch konvertiert wurden.
Oracle DW	Amazon Redshift	Unterstützung der automatischen Tabellenpartitionierung für die Datenmigration implementiert. Um die Datenmigration zu beschleunigen, AWS SCT können große Tabellen oder Partitionen automatisch auf der Grundlage der Werte in der `ROWID` Pseudospalte partitioniert werden. Weitere Informationen finden Sie unter Native Partitionierung verwenden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Unterstützung nativer `MERGE` Befehle im konvertierten Amazon Redshift Redshift-Code wurde implementiert. Weitere Informationen zu dem `MERGE` Befehl in Amazon Redshift finden Sie unter MERGE im Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	Die Konvertierung von `DELETE UPDATE` AND-Anweisungen, die keine expliziten Tabellennamen verwenden, wurde verbessert.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem `IN NOT IN` Anweisungen falsch konvertiert wurden.

Versionshinweise für AWS SCT Build 670

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurden Probleme behoben, bei denen das Aktionselement 9996 während der Konvertierung der folgenden Codeelemente unerwartet angezeigt wurde. • `CREATE INDEX`Aussagen innerhalb `INCLUDE` von Anweisungen • `DECLARE`-Anweisungen. • `DECLARE ... TABLE`-Anweisungen. • `DECLARE`mit Standardwerten innerhalb von `LOOP` Anweisungen • `DELETE`-Anweisungen. • `DROP CONSTRAINT` Aussagen innerhalb von `ALTER TABLE` Aussagen • `EXECUTE AS CALLER` und `REVERT` • `IIF`-Anweisungen. • Listen von Ausdrücken • `MONTH()`-Funktionen • `UPDATE`-Anweisungen. • `YEAR()`-Funktionen

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Unterstützung für Azure Synapse Analytics als Quelle für den Multiserver-Assessment-Prozess hinzugefügt. Weitere Informationen finden Sie unter Bewertungsbericht für mehrere Server.
Hadoop	Amazon EMR	Unterstützung für die Migration von Hadoop-Clustern zu Amazon EMR im Befehlszeilenschnittstellenmodus (CLI) implementiert. Weitere Informationen finden Sie unter Migration von Big-Data-Frameworks.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Resolver-Fehler behoben, der für Quelltabellen und -spalten auftrat.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `CASE` Ausdrücken wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `CURRENT_DATE` Verweisen auf spezielle Register wurde implementiert. Ein Verweis auf ein spezielles Register in Db2 for z/OS ist ein Verweis auf einen Wert, der vom aktuellen Server bereitgestellt wird.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `DATE` und `POSSTR` Funktionen wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Datetime-Konstanten wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Standardwerten für Spalten der folgenden Datentypen wurde verbessert:`DATE`, `TIMETIMESTAMP` , und. `TIMESTAMP WITH TIME ZONE`
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Anweisungen angezeigt wurde. `SELECT INTO`
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `DATEDIFF` Funktionen wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem `ISNULL` Funktionen konvertiert wurden`NULLIF`. Infolgedessen führte der konvertierte Code zu anderen Ergebnissen als der Quellcode. AWS SCT Konvertiert jetzt `ISNULL` Funktionen in`COALESCE`.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Verbesserte Datenextraktionsagenten zur Behebung eines Problems, bei dem für erfolgreich abgeschlossene Aufgaben der Status „Fehlgeschlagen“ gesetzt wurde.
Netezza	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, Endpunkte in Unteraufgaben nach dem Start einer Datenmigration mit Datenextraktionsagenten zu ändern.
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Es wurde die Möglichkeit hinzugefügt, mithilfe eines IPv6 Adresspro tokolls eine Verbindung zu Datenbanken herzustellen.
Oracle	Amazon RDS für Oracle	Die Konvertierung des `DBMS_JOB` Pakets, das Jobs in der Job-Warte schlange plant und verwaltet, wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Dem Erweiterungspaket wurden neue Funktionen hinzugefügt, um die Konvertierung der globalen verschachtelten Tabellen zu verbessern. Diese neuen Funktionen emulieren `DELETEEXTEND`, und `TRIM` Funktionen in Ihrem Oracle-Quellcode.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit hinzugefügt, den Konvertierungsbereich für SQL-Code anzugeben, der in Java-Anwendungen eingebettet ist. Sie können jetzt die Teilmengen des Quellanwendungsprojekts vom Konvertierungsumfang ausschließen. Weitere Informationen finden Sie unter Konvertieren Sie den SQL-Code Ihrer Java-Anwendung in AWS SCT.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Verkettungsoperatoren (`||`) innerhalb von Funktionsindizes wurde verbessert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `IN` Bedingungen, bei denen Ihr Quellcode keine Klammern für einen einzelnen Ausdruck enthält.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `MERGE` Anweisungen `INSERT ON CONFLICT` in PostgreSQL.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Parsing-Fehler behoben, der bei Prozedurpaketen auftrat.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 5072 während der Konvertierung von Paketen unerwartet angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Es wurde ein Fehler behoben, bei dem das Erweiterungspaket AWS SCT nicht angewendet wurde, wenn der konvertierte Code auf die Zieldatenbank angewendet wurde.
Oracle DW	Amazon Redshift	Es wurde ein Fehler behoben, bei dem einige der Erweiterungspaket-Dateien AWS SCT nicht angewendet wurden, wenn der Erweiterungspaket-Assistent verwendet wurde.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenmigration nicht verarbeitet werden AWS SCT konnte, AWS Snowball Edge wenn mehr als 500 Aufgaben parallel ausgeführt wurden.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem benutzerdefinierte Funktionen mit benutzerdefinierten Typen falsch konvertiert wurden.

Versionshinweise für Build 669 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Der Bewertungsprozess für mehrere Server wurde verbessert, mit dessen Hilfe die optimale Zieldatenbankplattform für Ihre Quelldatenbanken ermittelt werden kann. AWS SCT Ignoriert jetzt den AWS Secrets Manager Schlüssel, wenn Sie Datenbankanmeldeinformationen in der Eingabedatei mit kommagetrennten Werten (CSV) angeben. Weitere Informationen finden Sie unter Bewertungsbericht für mehrere Server.
Alle	Alle	Es wurde ein Problem behoben, bei dem der Multiserver-Assessmentbericht die IP-Adresse Ihrer Quelldatenbank enthielt, wenn für die Verbindung mit der Datenbank ein geheimes AWS Secrets Manager Formular verwendet wurde.
Alle	Amazon Redshift	Die automatische Konfiguration der Einstellungen für die Java Virtual Machine (JVM) wurde je nach Betriebssystem und verfügbarem RAM

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		implementiert. AWS SCT verwendet diese JVM, um die Arbeit der Datenextraktionsagenten auszuführen.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsag enten in Ubuntu nicht gestartet wurden.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsau fgaben nach dem Ausführen der `StartAgent.bat` Datei in Windows nicht gestartet wurden.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQ L PostgreSQ L	Es wurde ein Problem behoben, bei dem Spaltennamen falsch konvertiert wurden, wenn die Option „Eindeutige Namen für Indizes generieren“ aktiviert war.
Greenplum	Amazon Redshift	Die Konvertierung von Funktionen, die `VOID` zu Prozeduren zurückkehren, wurde implementiert.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenmigration fehlschlug, wenn die Quelldatenbank keine numerischen Werte (NaN) in numerischen Spalten enthielt. AWS SCT Datenextr aktionsagenten ersetzen jetzt NaN-Werte durch NULL.
IBM Db2 für z/OS	Aurora PostgreSQ L PostgreSQ L	Es wurde eine neue Konvertierungseinstellung hinzugefügt, um die `TIME FORMAT` Optionen `DATE FORMAT` und während der Konvertie rung `CHAR` integrierter Funktionen festzulegen.
IBM Db2 für z/OS	Aurora PostgreSQ L PostgreSQ L	Es wurde ein Aktionselement 8534 für die Konvertierung von vordefinierten Cursorn hinzugefügt, die mit der Klausel deklariert wurden. `WITHOUT RETURN` Wenn Ihr Cursor keine Ergebnismengen zurückgibt, AWS SCT weist er Ihrem Cursornamen im konvertierten Code einen `NULL` Wert zu und löst ein Aktionselement aus.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die `CURRENT CLIENT_APPLNAME` Eigenschaft, die AWS SCT während der Verbindung zur Quelldatenbank identifiziert wurde, wurde bearbeitet.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde eine neue Konvertierungseinstellung implementiert, um die `TIME FORMAT` Optionen `DATE FORMAT` und bei der Konvertierung `CHAR` integrierter Funktionen festzulegen.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `LEAVE` Anweisungen in `BEGIN...END` Blockanweisungen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von`XMLPARSE`, `XMLTABLE` und `XMLNAMESPACES` -Funktionen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung der `CHAR` integrierten Funktionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Cursorn.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von `F0R` Loop-Anweisungen angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung der in `SELECT` Anweisungen verwendeten Tabellentypen wurde verbessert.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version 2.2.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
Netezza	Amazon Redshift	Verbesserte Datenextraktionsagenten zur Behebung eines Problems, bei dem eine Zeile während der laufenden Datenreplikation nicht aus der Zieltabelle gelöscht wurde.
Oracle	Amazon RDS für Oracle	Verbesserte Konvertierung der Funktionen der Oracle Database Enterprise Edition.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `GROUPING_ID` Funktionen wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Die SQL-Codekonvertierung in C#-Anwendungen wurde verbessert, indem Unterstützung für die Zuordnung benutzerdefinierter Datentypen im Befehlszeilenschnittstellenmodus (CLI) hinzugefügt wurde.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung verschachtelter Tabellen wurde verbessert, um ein unerwartetes Aktionselement 9996 zu vermeiden.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem der Aufruf eines Objektkonstruktors falsch konvertiert wurde.
Oracle DW	Amazon Redshift	Unterstützung vorhandener Tabellenpartitionen für die Datenmigration implementiert. Um die Datenmigration zu beschleunigen, AWS SCT erstellt Unteraufgaben für jede Partition der Quelltabelle, die nicht leer ist. Weitere Informationen finden Sie unter Native Partitionierung verwenden.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `CAST` Funktionen mit `TIMESTAMP AS TIME WITH TIME ZONE` Argumenten `TIME WITH TIME ZONE AS TIMESTAMP TIME WITH TIME ZONE AS CHAR`, und.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung von `CAST` Funktionen mit der `FORMAT` Option.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem `CEIL` Funktionen nicht konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem `MERGE` Aussagen mit `DELETE` Klauseln falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem `TO_CHAR` Funktionen mit Datums- und Formatargumenten falsch konvertiert wurden.

Versionshinweise für AWS SCT Build 668

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem Multiplikationsoperatoren in Migrationsregeln nicht richtig funktionierten. Diese Operatore n ermöglichen es, die Länge von`char`, `varcharnvarchar`, und `string` Datentypen zu ändern. Weitere Informationen finden Sie unter Migrationsregeln erstellen.
Azure Synapse Analytics	Amazon Redshift	Unterstützung von `CONVERT` Funktionen mit `VARCHAR` Argumenten implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `SELECT` Aussagen mit `NOLOCK` Klauseln.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `UPDATE` Aussagen mit Aliasnamen oder mit `SET` UND-Klauseln`FROM`.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Automatische virtuelle Partitionierung für die Datenmigration implementiert. AWS SCT verwendet die `GP_SEGMENT_ID` Systemspalte, um Partitionen zu erstellen.
Greenplum	Amazon Redshift	Unterstützung von `RETURN QUERY` `RETURN SETOF` UND-Klauseln implementiert.
Greenplum	Amazon Redshift	Unterstützung von `SUBSTRING` Funktionen mit drei Parametern implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `SUBSTR` Funktionen mit `LOCATE` Parametern.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde eine Option hinzugefügt, um ein Array von `REFCURSOR` Variablen zur Rückgabe dynamischer Ergebnismengen zu verwenden. Wenn Sie diese Option in den Konvertierungseinstellungen auswählen, wird dem konvertierten Code ein zusätzlicher `OUT` Parameter AWS SCT hinzugefügt.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Unterstützung von `FOR` Loop-Anweisungen implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Unterstützung von `XMLPARSE` Funktionen implementiert. Es wurde ein Aktionspunkt 8541 für die Markierung von Leerzeichen in `XMLPARSE` Funktionen hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung mehrerer Ausnahmebehandler in einem einzigen Block wurde verbessert. `BEGIN ... END`
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `INSERT` und `DELETE` Triggern wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von verschachtelten Prozeduraufrufen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Tabellentypen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem bitweise logische `NOT` Operationen fälschlicherweise in Ganzzahlwerte konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem lokale Arrays in PostgreSQL Version 8.0.2 und niedriger nicht initialisiert wurden.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem `MERGE` Aussagen mit `WHEN NOT MATCHED BY SOURCE` Klauseln falsch konvertiert wurden.
MySQL	Aurora MySQL	Es wurde ein Problem behoben, bei dem die Benutzerberechtigungen, die von der `rds_superuser_role` Rolle gewährt wurden, AWS SCT falsch bestimmt wurden.
Netezza	Amazon Redshift	Das Ladeprogramm für Quellmetadaten wurde verbessert, um sicherzustellen, dass Datenbankobjekte mit Namen in Kleinbuchstaben AWS SCT korrekt geladen werden.
Oracle	Aurora PostgreSQL PostgreSQL	Dem Erweiterungspaket wurden neue Funktionen hinzugefügt, um die Konvertierung von lokalen verschachtelten Tabellen zu verbessern. Diese neuen Funktionen emulieren `PRIOR`,,`NEXT`,`LIMIT`,`FIRST`,`LAST`,`EXISTS`, `EXTEND` `TRIMDELETE`, und `SET` Funktionen in Ihrem Oracle-Quellcode. Weitere Informationen finden Sie unter Erweiterungspakete.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit hinzugefügt, den Konvertierungsbereich für C#-Anwendungen anzugeben. Benutzer können jetzt die Teilmengen des Quellanwendungsprojekts vom Konvertierungsumfang ausschließen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Unterstützung von `COUNT` Methoden in Sammlungen implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Unterstützung von Variablen und Konstruktoren in verschachtelten Tabellen implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Unterstützung von `RATIO_TO_REPORT` und `STANDARD_HASH` Funktionen implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung großer Objekte (LOBs) als Teil des AWS SCT Erweiterungspakets.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung lokaler Sammlungen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `JOIN` Anweisungen mit `USING` Klauseln, bei denen Spaltennamen den Tabellennamen nicht enthalten.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `EMPTY_BLOB` und `EMPTY_CLOB` Funktionen wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Positionsbindungsvariablen in C#-Anwendungen wurde implementiert.
SAP ASE	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Triggern mit mehreren Ereignissen wurde implementiert.
SAP ASE	Aurora PostgreSQL PostgreSQL	Die Konvertierung von rekursiven Triggern wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Triggern mit der `@@rowcount` globalen Variablen.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Aggregatfunktionen in der `SET` Klausel von `UPDATE` Anweisungen falsch konvertiert wurden.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 42702 während der Konvertierung von `UPDATE` Anweisungen unerwartet angezeigt wurde.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem `CONVERT` Funktionen mit `CHAR` Argumenten falsch konvertiert wurden.
Snowflake	Amazon Redshift	Unterstützung für Snowflake als Quelle für Datenmigration mit AWS SCT Datenextraktionsagenten hinzugefügt. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `CAST` Funktionen mit `TIMESTAMP AS TIME WITH TIMEZONE` Argumenten.

Versionshinweise für AWS SCT Build 667

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung von Informatica-Skripts zum Extrahieren, Transform ieren und Laden (ETL) im CLI (Command Line Interface) -Modus implementiert. AWS SCT leitet Ihre Informatica ETL-Skripts automatis ch zur neuen Zieldatenbank um. AWS SCT Konvertiert außerdem Objektnamen und SQL-Code, der in Ihre Informatica-Objekte eingebettet ist. Weitere Informationen finden Sie unter Informatica ETL-Skripts.
Alle	Amazon Redshift	Die unterstützte Mindesttreiberversion für Amazon Redshift wurde auf 2.1.0.9 erhöht. Weitere Informationen finden Sie unter Installation von JDBC-Treibern für AWS Schema Conversion Tool.
Azure Synapse Analytics	Amazon Redshift	Dem Erweiterungspaket wurde eine neue Funktion hinzugefügt, um die Konvertierung der `CONVERT` Funktion mit drei Datums- und Uhrzeitargumenten zu verbessern.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung der `DATEDIFF` Funktion.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Die Version des Erweiterungspakets wurde aktualisiert. Stellen Sie sicher, dass Sie die neueste Version des Erweiterungspakets in Ihren bestehenden AWS SCT Projekten anwenden. Weitere Informationen finden Sie unter Erweiterungspakete.
BigQuery	Amazon Redshift	Es wurde ein Problem behoben, bei dem die gefilterten Objekte im Befehlszeilenschnittstellenmodus (CLI) nicht konvertiert wurden.
Greenplum	Amazon Redshift	Es wurde ein Fehler behoben, bei dem temporäre Tabellen, die in einer gespeicherten Prozedur deklariert wurden, AWS SCT nicht konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Spaltencodierungsattribute im konvertierten Code fehlten.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `UPDATE` Anweisungen für selbstreferenziere nde Tabellen mit mehr als einer `INNER JOIN` Klausel wurde implementiert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Unterstützung für `deleted` temporäre Tabellen, die `inserted` SQL Server für DML-Trigger verwendet, wurde implementiert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von benutzerdefinierten Typen in gespeicherten Prozeduren, die in verschiedenen Datenbankschemas erstellt wurden. Es wurde ein Problem behoben, bei dem der Datentyp nicht gefunden werden AWS SCT konnte und der Aktionspu nkt 9996 angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem im konvertierten Code unerwartet eckige Klammern (`[ ]`) um die Namen der Datenbank objekte standen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem `@@ROWCOUNT` Funktionen falsch konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW	Amazon Redshift	Unterstützung von `geometry` `geography` Datentypen wurde implementiert.
Microsoft SQL Server DW	Amazon Redshift	Die Unterstützung des `MAX` Schlüsselworts in Datentypdeklarationen im konvertierten Code wurde implementiert.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung von `DATEADD` Funktionen.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Code-Konvertierung in Java-Anwendungen durch Hinzufügen von Unterstützung für das MyBatis Framework. Weitere Informationen finden Sie unter SQL-Code in Java.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Codekonvertierung in Java-Anwendungen, die das MyBatis Framework verwenden. Es wurde ein Aktionselement 30411 für SQL-Code mit nicht unterstützter Syntax hinzugefügt.
Oracle	Aurora PostgreSQL PostgreSQL	Die SQL-Codekonvertierung in Pro*C-Anwendungen wurde verbessert, indem Unterstützung für Deklarationen hinzugefügt wurde. `typedef struct`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Unterstützung von UND-Anweisungen `CROSS JOIN` implementiert. `LEFT JOIN`
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `MERGE` Aussagen. Es wurde ein Problem behoben, bei dem einzufügende Werte im konvertierten Code fehlten.
Teradata	Amazon Redshift	Die standardmäßigen Codierungseinstellungen für die Spaltenkomprimierung, die im konvertierten Code AWS SCT verwendet werden, wurden so geändert, dass sie den Standardeinstellungen von Amazon Redshift entsprechen. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem mathematische Operationen, die den `TIME` Datentyp verwenden, falsch konvertiert wurden.
Teradata	Amazon-Redshift-RSQL	Die Konvertierung von FastExport Code, der sich in Shell-Skripten befindet, wurde implementiert.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, bei dem Anweisungen AWS SCT nicht konvertiert wurden. `COALESCE %data`
Vertica	Amazon Redshift	Die Vorschläge zur Konversionsoptimierung wurden verbessert, wenn ein Benutzer eine Optimierungsstrategie auswählt.

Versionshinweise für AWS SCT Build 666

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Analysefehler behoben, der bei `ON` Klauseln auftrat, die sich in `JOIN` Anweisungen befinden.
Azure Synapse Analytics	Amazon Redshift	Dem Erweiterungspaket wurden drei neue Funktionen hinzugefügt, um die Konvertierung der `CONVERT` Funktion mit Datums- und Uhrzeitargumenten zu verbessern.
Azure Synapse Analytics	Amazon Redshift	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Systemdatenbankschemas AWS SCT geladen werden.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Resolver-Fehler behoben, der bei Spalten temporärer Tabellen auftrat.
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von `BINARY VARBINARY` Datentypen in den `VARBYTE` Datentyp wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Die Unterstützung des `TIME` Datentyps im konvertierten Code wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `COLLATE` Klauseln. Es wurde ein Problem behoben, bei dem das Aktionselement 31141 bei der Konvertierung von Spalten mit der Standardsortierung der Datenbank unerwartet angezeigt wurde.
BigQuery	Amazon Redshift	Die Konvertierung von Prozeduren, die Eingabeparameter ändern, wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Abfrage AWS SCT verwendet wurde, die nicht mit Greenplum 6.x-Datenbanken kompatibel ist.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `EXCEPTION` Abschnitten durch Übertragung von Ausnahmehandlern von Db2 for z/OS nach PostgreSQL.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Kontoauszügen. `OPEN CURSOR`
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `IIF` Funktionen mithilfe von `CASE` Ausdrücken wurde implementiert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Prozeduren mit Tabellenwertparametern falsch konvertiert wurden, wenn die `CREATE PROCEDURE` Anweisung keinen `BEGIN...END` Block enthielt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem die `SCOPE_IDENTITY` Funktion falsch konvertiert wurde.
Oracle	Amazon RDS für Oracle	Es wurde ein Loader-Fehler behoben, der bei der `SELECT_CATALOG_ROLE` Rolle auftrat, wenn Oracle 10g als Quelle verwendet wurde.
Oracle	Amazon RDS für Oracle	Der Loader wurde verbessert, sodass er Oracle Scheduler-Jobs unterstützt.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `JOIN` Anweisungen mit `USING` Klauseln wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Leistung des konvertierten Codes, wenn der Quellcode globale Variablen in `WHERE` Klauseln enthält.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Code-Konvertierung in Java-Anwendungen durch Hinzufügen von Unterstützung für das MyBatis Framework. Weitere Informationen finden Sie unter SQL-Code in Java.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Die Konvertierung der Operatoren `PIVOT` und der `UNPIVOT` relationalen Operatoren wurde implementiert.
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Quellcode, der JSON-Objekte verwendet, nicht konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die von einem gelöschten Benutzer erstellten Tabellen nicht korrekt geladen wurden.
Teradata	Amazon Redshift	Die Konvertierung von `INSTR` Funktionen in native Amazon Redshift `STRPOS` Redshift-Funktionen wurde implementiert.
Teradata	Amazon Redshift	Die Konvertierung der `TRANSLATE` Funktionen `NVP` und wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `COALESCE` Ausdrücken.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `DECLARE CONDITION` Kontoauszügen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `EXTRACT` Funktionen mit dem `SECOND` Syntaxelement.
Teradata	Amazon Redshift	Verbesserte Konvertierung der `SQLCODE` Variablen `SQLSTATE` und in den `LOOP` Anweisungen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von eindeutigen Indizes.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet bei der Konvertierung von `CURRENT_TIMESTAMP` Anweisungen angezeigt wurde, deren Genauigkeit auf 3 gesetzt war.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Backslashes fälschlicherweise in Zeichenfolgenliterale umgewandelt wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte `EXEC` Anweisungen einen falschen Feldnamen in der Anweisung enthielten. `ADD CONSTRAINT`
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte `QUALIFY` Unterabfragen einen falschen Unterabfragenamen enthielten.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte Ansichten nicht angewendet wurden. Es wurde eine explizite Umwandlung in einen bestimmten Datentyp für NULL-Werte im konvertierten Code hinzugefügt.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Datums- und Uhrzeitfu nktionen falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem hexadezimale Zeichenke ttenliterale nicht konvertiert wurden.

Versionshinweise für Build 665 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von `CONCAT` Funktionen mit `VARCHAR` Argumenten wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `CREATE TABLE` Anweisungen, die temporäre Tabellen erstellen und den Schemanamen nicht enthalten. AWS SCT erstellt das dbo Schema zum Speichern dieser temporäre n Tabellen in der Zieldatenbank.
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von `DROP TABLE` Anweisungen, die Sie in temporären Tabellen ausführen, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `OBJECT_ID` Anweisungen mit den `BEGIN...END` Blöcken.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem gespeicherte Prozeduren mit Blockkommentaren nicht konvertiert werden AWS SCT konnten.
BigQuery	Amazon Redshift	Die Umstellung von BigQuery Data Warehouses auf Amazon Redshift wurde implementiert. Weitere Informationen finden Sie unter Verbindung zu Google herstellen BigQuery mit AWS Schema Conversion Tool.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Triggern, die mehrere Ereignisse verarbeiten `inserted` und mit `deleted` Systemtabellen in SQL Server arbeiten.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Resolver-Fehler behoben, der für `inserted` und `deleted` Systemtabellen in SQL Server auftrat.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version 2.1.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora MySQL MariaDB MySQL	Es wurde ein Problem behoben, bei dem der `varchar2` Datentyp falsch konvertiert wurde.
Oracle	Aurora MySQL Aurora PostgreSQL MariaDB MySQL PostgreSQL	AWS SCT Unterstützt für Oracle-Datenbanken ab Version 12c die folgenden erweiterten Datentypen: • `VARCHAR2` • `NVARCHAR2` • `RAW` AWS SCT Die maximal unterstützte Spaltenlänge für diese Datentypen wurde von 8.000 auf 32.767 Byte erhöht.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Analysefehler behoben, der für das Oracle Event Processing Package aufgetreten ist.
Teradata	Amazon Redshift	Es wurde ein Aktionspunkt 13214 für mehrere `RESET WHEN` Klauseln in einer einzigen Anweisung hinzugefügt. `SELECT`
Teradata	Amazon Redshift	Es wurde ein Aktionselement für `SQLSTATE` Variablen hinzugefügt, die sich außerhalb eines Ausnahmebehandlungsblocks befinden.
Teradata	Amazon Redshift	Die Konvertierung von `ACTIVITY_COUNT` Variablen in wurde implementiert`ROW_COUNT` .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Konvertierung der integrierten `ST_TRANSFORM` Geometrie funktion wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von Löschanweisungen in Ansichten ohne die `WHERE` Klausel.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `CAST` Operatoren in Ausdrücken.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `GROUP BY` Klauseln.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `INSTR` und `REGEXP_INSTR` integrierten Funktionen.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Aliasverweise auf seitliche Spalten falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Spaltennamen in der `QUALIFY` Unterabfrage falsch konvertiert wurden.
Teradata	Amazon Redshift	Die Konvertierung von `.QUIT` Befehlen mit dem Schlüsselwort `ERRORCODE` status value wurde implementiert.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 während der Konvertierung von Anweisungen unerwartet angezeigt wurde. `CREATE`
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 während der Konvertierung von Anweisungen unerwartet angezeigt wurde. `END`

Versionshinweise für AWS SCT Build 664

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für Amazon Redshift Serverless als Quelle und Ziel für Datenbankmigrationsprojekte in hinzugefügt. AWS SCT Um eine Verbindung zu Amazon Redshift Serverless herzustellen, stellen Sie sicher, dass Sie den Amazon Redshift JDBC-Treiber Version 2.1.0.9 oder höher verwenden.
Alle	Alle	Die Benutzeroberfläche des Fensters mit den Konvertierungseins tellungen wurde verbessert. AWS SCT zeigt jetzt nur Einstellungen für Datenbankkonvertierungspaare mit erstellten Zuordnungsregeln an. Weitere Informationen finden Sie unter Zuordnung von Datentype n.
Alle	Alle	Der Bewertungsbericht wurde aktualisiert, sodass doppelte Informati onen über die Linie und Position des Aktionspunkts entfernt wurden.
Alle	Amazon Redshift	Automatischer Speicherausgleich bei Datenextraktionsaufgaben implementiert.
Alle	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Datenextraktionsag enten keine Verbindung zu AWS Snowball Edge Geräten herstellen konnten.
Azure SQL Database IBM Db2 für z/OS IBM Db2 (LUW) Microsoft SQL Server	Aurora MySQL Aurora PostgreSQ L MariaDB MySQL PostgreSQ L	Die Unterstützung von SUSE Linux 15.3 als Plattform für die Ausführung von Datenextraktionsagenten wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
MySQL Oracle PostgreSQL SAP ASE		
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `DATEADD` Funktionen.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit hinzugefügt, die Spaltensortierung in Migrationsregeln zu ändern.
Microsoft SSIS	AWS Glue AWS Glue Studio	Es wurde ein unerwarteter Fehler behoben, der auftrat, wenn Benutzer ein Quellskript auswählten.
Oracle	Aurora MySQL MariaDB MySQL	Die Konvertierung der Verwendung von gespeicherten Funktionen als generierte Spaltenausdrücke wurde implementiert. AWS SCT erzeugt Trigger, um dieses Verhalten zu emulieren, da MySQL die Verwendung von gespeicherten Funktionen als generierte Spaltenausdrücke nicht unterstützt.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Funktionen aus dem `UTL_MATCH` Paket wurde als Teil des AWS SCT Erweiterungspakets implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung der `REGEXP_LIKE` Funktion mit dem `NULL` Parameter wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung der `SYS_EXTRACT_UTC` Funktion.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Codekonvertierung in C++-Anwendungen durch Implementierung der Unterstützung von `WcscatsWcscpys`, und `Wcsncats` Funktionen. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in C++-Anwendungen mit AWS Schema Conversion Tool.
Oracle DW Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte Anweisungen keine explizite Konvertierung von Werten in den Spaltendatentyp enthielten. Dieses Problem trat bei Anweisungen auf, die Abfrageergebnisse aus anderen Tabellen verwenden.
Teradata	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, die Spaltensortierung zwischen `case sensitive` und `case insensitive` in Migrationsregeln zu ändern. Weitere Informationen finden Sie unter Anwendung von Migrationsregeln.
Teradata	Amazon Redshift	Es wurde ein Resolver-Fehler behoben, der bei Anweisungen auftrat. `CREATE TABLE AS`
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die integrierte `P_INTERSECT` Funktion mit einem `COALESCE` Ausdruck nicht konvertiert wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Konvertierung von Spalten mit dem Namen `OID` nach wurde implementiert`_OID`, um die Verwendung eines reservierten Schlüssel worts in Amazon Redshift zu vermeiden.
Teradata	Amazon Redshift	Die Konvertierung von `RENAME` Anweisungen für Funktionen, Prozeduren, Ansichten und Makros wurde implementiert.
Teradata	Amazon Redshift	Die Konvertierung der `STROKE` Funktion zur `SPLIT_PART` Funktion in Amazon Redshift wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung der Funktionen `INSTR` und des `REGEXP_INSTR` Systems.
Teradata	Amazon Redshift	Verbesserte Konvertierung des `TIME` Datentyps.
Teradata	Amazon Redshift	Verbesserte Emulation der `SET` `MULTISET` UND-Tabellen durch Implementierung der Konvertierung primärer und sekundärer eindeutiger Indizes.
Teradata	Amazon Redshift	Es wurde ein Analysefehler behoben, der bei der Funktion aufgetret en ist. `CHARACTER`
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, der auftrat, als Benutzer Teradata Basic Teradata Query-Skripts (BTEQ) aus dem Projekt entfernten. AWS SCT

AWS SCT Versionshinweise für Build 663

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, die Länge von`char`, `varcharnvarchar`, und `string` Datentypen mithilfe des Multiplik ationsoperators in einer Migrationsregel zu ändern. Weitere Informati onen finden Sie unter Anwendung von Migrationsregeln.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Die Unterstützung von drei neuen Spalten im Multiserver-Assessment-Bericht wurde implementiert und das Format der Eingabedatei aktualisiert. Stellen Sie sicher, dass Sie die aktualisierte Vorlage der Eingabedatei mit der neuesten Version von AWS SCT verwenden. Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen in AWS Schema Conversion Tool.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von `OBJECT_ID` Kontoauszügen.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Unterstützung für Babelfish for Aurora PostgreSQL 1.2.0 als Zielplattform für Bewertungsberichte zur Datenbankmigration hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Funktionen in Babelfish nach Version im Amazon Aurora Aurora-Benutzerhandbuch.
Microsoft SQL Server DW	Amazon Redshift	Unterstützung für `AT TIME ZONE` Klauseln hinzugefügt.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Anweisung außerhalb des `BEGIN/END` Blocks falsch konvertiert wurde.
Netezza	Amazon Redshift	Die Konvertierung des `TIME` Datentyps wurde verbessert und die Konvertierung verwandter integrierter Funktionen, Ausdrücke und Literale implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Loader-Fehler behoben, der auftrat, wenn Oracle 10g als Quelle verwendet wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von `OFFSET FETCH` UND-Klauseln.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Prozeduren mit `OUT` Parametern mit Standardwerten falsch konvertiert wurden.
Oracle DW	Amazon Redshift	Verbesserte Konvertierung von Oracle-Funktionen in benutzerdefinierte Funktionen von Amazon Redshift.
Snowflake	Amazon Redshift	Verbesserte Konvertierung von Klauseln. `WITH`
Teradata	Amazon Redshift	Es wurde ein neues Aktionselement 13209 für nicht unterstützte Multibyte-Zeichen für den Datentyp hinzugefügt. `CHAR`
Teradata	Amazon Redshift	Es wurde ein Loader-Fehler behoben, bei dem die Tabellen nicht vollständig geladen wurden.
Teradata	Amazon Redshift	Es wurde ein Transformer-Fehler behoben, bei dem die eingebaute `P_INTERSECT` Funktion in einer `JOIN` Bedingung nicht konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem der Name einer Ansicht falsch konvertiert wurde, wenn die `SELECT` Anweisung für eine Tabelle mit Sonderzeichen im Namen ausgeführt wurde.
Teradata	Amazon Redshift	Die Konvertierung von `INSERT` Anweisungen mit dem `UNTIL_CHANGED` Wert im `PERIOD(DATE)` Datentyp wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung der integrierten `FORMAT` Funktion mithilfe der `TO_CHAR` Funktion in Amazon Redshift.
Teradata	Amazon Redshift	Die Konvertierung der integrierten `RANK` Funktion wurde verbesser t, um sicherzustellen, dass der konvertierte Code NULL-Werte in derselben Reihenfolge wie der Quellcode zurückgibt.
Teradata	Amazon Redshift	Verbesserte Konvertierung von eindeutigen Einschränkungen wie primären oder sekundären eindeutigen Indizes.

Versionshinweise für AWS SCT Build 662

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, bei der Erstellung des Multiserver-Assessment-Berichts automatisch AWS SCT Projekte für jede Quelldatenbank zu erstellen. Wenn diese Option aktiviert ist, AWS SCT können Sie diesen Projekten Zuordnungsregeln hinzufüge n und Konvertierungsstatistiken für die Offline-Verwendung speichern . Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen in AWS Schema Conversion Tool.
Alle	Alle	Die Unterstützung von Prozent (%) als Platzhalter in Datenbank- und Schemanamen bei der Erstellung des Multiserver-Assessment-Beri chts wurde implementiert.
Alle	Aurora MySQL Aurora PostgreSQ L	Die Laufzeit aller AWS Lambda Funktionen wurde auf Python-Ve rsion 3.9 aktualisiert.
Alle	Amazon Redshift	Alle Datenextraktionsagenten wurden für die Verwendung aktualisiert AWS SDK for Java 2.x.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `DELETE` Aussagen mit `NON EXISTS` Klauseln wurde verbessert.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Verbindung zu einer Quelldatenbank fehlschlug.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem der konvertierte Code eines Triggers zwei Erwähnungen des Objektalias enthielt.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Objekten mit Namen in gemischter Groß- und Kleinschreibung wurde verbessert, wenn die Option Datenbank objektnamen mit Groß- und Kleinschreibung behandeln aktiviert ist.
Microsoft SQL Server DW Teradata	Amazon Redshift	Die Konvertierung der Operatoren `PIVOT` und der `UNPIVOT` relationa len Operatoren wurde implementiert.
Netezza	Amazon Redshift	Die Konvertierung des `TIME` Datentyps wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Die konstante Konvertierung des `UTL_TCP.CRLF` Pakets wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem mit Erweiterungspaketen behoben, bei dem die Länge von Datentypen für Spalten mit variabler Länge während der Konvertierung nicht beibehalten wurde.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von SQL-Code in C++-Anwendungen wurde implementiert. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in C++-Anwendungen mit AWS Schema Conversion Tool.
Oracle	Aurora PostgreSQL PostgreSQL	Unterstützung für die Benennung von Groß- und Kleinschreibung für die Konvertierung von globalen Variablen und assoziativen Arrays implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung der `TO_NUMBER` Funktionen `TO_CHAR` `TO_DATE`, und im Erweiterungspaket wurde verbessert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung des `TABLE()` Operators.
Oracle DW	Amazon Redshift	Unterstützung für die Konvertierung von Primärschlüsseln und andere Einschränkungen hinzugefügt.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 12054 bei der Konvertierung von bedingten Anweisungen nicht angezeigt wurde.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem während der Konvertierung von Tabellen mit Spalten benutzerdefinierten Typs ein Objekt mit einem leeren Namen im Zielbaum erstellt wurde.
SAP ASE	Aurora PostgreSQL PostgreSQL	Ein Loader-Fehler für gespeicherte Objekte wie Skripten, Routinen usw. wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem der Aktionspunkt 22152 bei Bedarf nicht angezeigt wurde und das Konvertierungsergebnis als Kommentar AWS SCT angezeigt wurde.
Snowflake	Amazon Redshift	Verbesserte Konvertierung der Datums- und Uhrzeitfunktionen; Unterstützung von Zeitzonen implementiert.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem nicht-rekursive allgemein e Tabellenausdrücke (CTEs) mit einer `WITH` Klausel als rekursiv konvertiert wurden. CTEs
Teradata	Amazon Redshift	Die Konvertierung von `UPDATE` Anweisungen mit Tabellenverknüpfun gen als Bedingung wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `RENAME TABLE` Kontoauszügen.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem leere Spalten in der Datei mit kommagetrennten Werten (CSV) mit einem Bewertungsbericht auftauchten.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, bei dem am Ende des konvertierten BTEQ-Makros (Basic Teradata Query) ein Semikolon fehlte.
Teradata	Amazon-Redshift-RSQL	Die Konvertierung mehrerer Datentypwerte in Anweisungen wurde verbessert. `CASE`
Teradata	Amazon-Redshift-RSQL	Verbesserte Konvertierung der `LIKE ANY` Klausel mit einem `ESCAPE` Zeichen.
Teradata	Amazon-Redshift-RSQL	Verbesserte Konvertierung der `CAST` Funktion in `INSERT` Anweisung en.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Verbesserte Konvertierung der Zeitzonen, Implementierung der Zuordnung von Zeitzonenregionen.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 unerwartet während der Konvertierung von Shell-Skripten mit BTEQ-Skripten angezeigt wurde.
Teradata	Amazon-Redshift-RSQL AWS Glue	Das Limit von 500 Zeichen für die Werte von Substitutionsvariablen wurde implementiert.
Vertica	Amazon Redshift	Die Konvertierung der `RAW` Datentypen `BINARYVARBINARY` ,`LONG BINARY`,`BYTEA`, und in den `VARBYTE` Datentyp wurde implementiert.
Vertica	Amazon Redshift	Verbesserte Konvertierung der integrierten Funktionen und Literale.

Versionshinweise für AWS SCT Build 661

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurden Filter hinzugefügt, um in der Mapping-Ansicht nach Zuordnungsregeln zu suchen. Wenn Sie einen Filter anwenden, AWS SCT werden Regeln angezeigt, die den Filterbedingungen in der Liste der Serverzuordnungen entsprechen. Weitere Informati onen finden Sie unter Bearbeiten von Datentypzuordnungen in AWS Schema Conversion Tool.
Alle	Alle	Apache Log4j wurde auf Version 2.17.1 aktualisiert.
Alle	Amazon Redshift	Unterstützung für die Datenmigration zu Amazon Redshift mithilfe der `ENCRYPTED` Klausel im `COPY` Befehl hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Die REST-API der Datenextraktionsagenten wurde verbessert. Die aktualisierte REST-API bietet Unterstützung für neue Eigenschaften wie Verschlüsselungsschlüssel, Verschlüsselungstyp usw.
Alle	Amazon Redshift	Die Übernahme einer Rolle in den Datenextraktionsagenten wurde implementiert. Dieses Update verbessert die Verteilung der Unteraufgaben und ermöglicht AWS SCT die Zuweisung von Aufgaben an freie Agenten mit der angegebenen Rolle.
Alle	Amazon Redshift	Es wurde eine Überprüfung implementiert, ob alle erforderlichen Komponenten installiert sind, bevor das Erweiterungspaket auf Amazon Redshift angewendet wird.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung der `ERROR_STATE` Systemfunktionen `ERROR_LINE` `ERROR_MESSAGE` `ERROR_NUMBER` ,`ERROR_PROCEDURE` ,`ERROR_SEVERITY` ,, und zur Fehlerbehandlung.
IBM Db2 für z/OS	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Unterstützung für IBM Db2 for z/OS Version 12 als Quelle für Datenbankmigrationsprojekte in hinzugefügt. AWS SCT Weitere Informationen finden Sie unter Verbindung zu IBM DB2 für z/OS herstellen.
IBM Db2 (LUW)	Alle	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Routineparameter AWS SCT geladen werden, die Spaltennamen duplizieren.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Ein Transformer-Fehler für Prozeduren mit der `SET NOCOUNT ON` Set-Anweisung wurde behoben.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung der `CONCAT` Funktion, wenn ein Eingabewert eine Variable vom benutzerdefinierten Typ ist.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem die `DATEPART` Funktion falsch konvertiert wurde.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version der Babelfish-Funktionskonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem Prozeduren mit einer `EXECUTE` Anweisung falsch konvertiert wurden.
Microsoft SSIS	AWS Glue	Die Benutzeroberfläche des Assistenten zur Jobkonfiguration wurde verbessert. AWS SCT zeigt jetzt nur verfügbare Verbindungen im Abschnitt Verbindungskonfiguration an.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SSIS	AWS Glue	Es wurde ein Problem behoben, bei dem die Transformationsregeln nicht auf Paketaufgaben und Variablenregeln angewendet wurden.
Microsoft SSIS	AWS Glue AWS Glue Studio	Es wurde ein neuer Aktionspunkt 25042 für nicht unterstützte Komponenten hinzugefügt.
Microsoft SSIS	AWS Glue Studio	Die Konvertierung von Microsoft SQL Server Integration Services (SSIS) -Paketen zum Extrahieren, Transformieren und Laden (ETL) in wurde implementiert. AWS Glue Studio Weitere Informationen finden Sie unter SSIS zu AWS Glue Studio.
Oracle	MariaDB	Ein Problem mit der Konvertierung des `MINUS` Operators wurde behoben.
Oracle	MariaDB	Verbesserte Konvertierung der `ADD_MONTHS` Funktione `nROWNUM`,`SYS_GUID`, und`TO_CHAR`, wenn die `sql_mode` Systemvar iable in MariaDB auf Oracle gesetzt ist.
Oracle	PostgreSQ L	Es wurde eine Option hinzugefügt, um die Konvertierung von Bind-Variablentypen in SQL-Typen in Konvertierungsprojekten für generische Anwendungen zu vermeiden.
Oracle	PostgreSQ L	Es wurde eine Option hinzugefügt, um zu verhindern, dass der Schemaname dem Namen des konvertierten Objekts in Konvertie rungsprojekten für generische Anwendungen hinzugefügt wird.
Oracle	PostgreSQ L	Unterstützung für das `?x` Bind-Variablenformat für die Konvertierung von Anwendungs-SQL-Code hinzugefügt.
Oracle DW	Amazon Redshift	Die Konvertierung des `RAW` Datentyps in den `VARBYTE` Datentyp wurde implementiert.
Teradata	Amazon Redshift	Es wurde eine Option hinzugefügt, um `SET` Tabellen im konvertierten Code zu emulieren. Für diese Emulation gibt es AWS SCT Unterstüt zungen `MIN` und `MAX` Bedingungen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Konvertierung von Join-Operationen mit Parametern verschied ener Datentypen wurde verbessert. Dieses Update ermöglicht AWS SCT die Anwendung von Transformationsregeln bei der Konvertie rung solcher Operationen.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die `GROUP BY` Klausel falsch konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die `QUALIFY` Klausel falsch konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein unerwarteter Fehler behoben, der beim Import von FastExport Skripten aufgetreten ist.
Teradata	Amazon-Redshift-RSQL	Es wurde die Möglichkeit implementiert, die Werte von Variablen in Teradata BTEQ- und Shell-Skripten zu bearbeiten.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Manifest-Skript für die konvertierten Teradata-Sitzungen fehlte. FastLoad
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Erweiterung der Manifestdatei im Uniform Resource Locator (URL) für die konvertie rten Skripts fehlte. FastLoad
Teradata BTEQ	Amazon-Redshift-RSQL	Ein Loader-Fehler für Skripte mit Substitutionsvariablen wurde behoben.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Aktionspunkt 27022 bei Bedarf nicht angezeigt wurde.

Versionshinweise für AWS SCT Build 660

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für AWS Secrets Manager und Secure Sockets Layer (SSL) wurde im Multiserver-Assessment-Bericht hinzugefügt. Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen in AWS Schema Conversion Tool.
Alle	Alle	Die Erfassung von Statistiken für konvertierte Objekte wurde verbessert.
Alle	PostgreSQL	Implementierte Unterstützung von PostgreSQL-Hauptversion 14 und MariaDB 10.6 als Migrationsziele.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Transformationslogik für die Namen konvertierter Objekte.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Verbesserte Konvertierung des `XML` Datentyps.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem `NOT LIKE` Klauseln falsch konvertiert wurden.
Microsoft Azure SQL Database	Aurora PostgreSQL	Ein Transformationsfehler für Prozeduren mit`INSERT`, und `UPDATE` -Anweisungen`DELETE`, die die `OUTPUT` Klausel enthalten, wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	PostgreSQL	
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Ein Transformationsfehler für Prozeduren mit der `RETURN @@ROWCOUNT` Anweisung wurde behoben.
Microsoft SQL Server	Alle	Die Konvertierung von Prozeduren, die Verbindungsserver verwenden, wurde verbessert.
Microsoft SQL Server	Alle	Unterstützung der Microsoft Windows-Authentifizierung im Multiserver-Assessment-Bericht hinzugefügt.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Ein Transformationsfehler für Tabellenwertkonstruktoren wurde behoben.
Microsoft SQL Server DW	Amazon Redshift und AWS Glue	Die Konvertierung von ETL-Skripts (Extrahieren, Transformieren und Laden) wurde verbessert, sodass nun der richtige Pfad zu den konvertierten Skripten angegeben wird.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem verschiedene konvertierte Skripts für virtuelle und reale Zieldatenbankplattformen generiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Unterstützung für die Konvertierung von Indizes für materialisierte Ansichten hinzugefügt.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 5982 bei der Konvertierung `PRIMARY KEY` und bei `UNIQUE` Einschränkungen mit dieser Option nicht angezeigt wurde. `NOVALIDATE`
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem zusätzliche Kategorien im konvertierten Schema angezeigt wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 13185 nicht angezeigt wurde, wenn eine ungelöste Spalte als Argument der `CAST` Funktion konvertiert wurde.
Teradata	Amazon Redshift	Die Konvertierung von `DELETE` `DELETE ALL` AND-Anweisungen zur Verwendung des `TRUNCATE` Befehls im konvertierten Code wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von `SET` Tabellen.
Teradata	Amazon Redshift	Verbesserte `NORMALIZE` Konditionsumwandlung.
Teradata	Amazon Redshift	Der Bewertungsbericht wurde aktualisiert, sodass die Statistiken zur Datenbankschemakonvertierung aus der Liste der Datenbank-Speicherobjekte entfernt wurden.
Teradata	Amazon Redshift	Die Konvertierung der `UPDATE` Anweisung ohne die `FROM` Klausel wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Unterstützung des `VARBYTE` Datentyps im konvertierten Code wurde implementiert.
Teradata BTEQ	AWS Glue	Es wurde ein Problem behoben, bei dem die AWS Glue Option „Konvertieren in“ im Kontextmenü deaktiviert war.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Datentypen im konvertierten Code fehlten.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Substitutionsvariablen im konvertierten Code falsch in Anführungszeichen gesetzt wurden.
Teradata BTEQ	Amazon-Redshift-RSQL	Ein Problem bei der Konvertierung von Substitutionsvariablen mit Werten in Skripten wurde behoben. FastLoad
Vertica	Amazon Redshift	Die Unterstützung des `TIME` Datentyps im konvertierten Code wurde implementiert.
Vertica	Amazon Redshift	Verbesserte Konvertierung von `ORDER BY` Ausdrücken `SELECT DISTINCT` und.
Vertica	Amazon Redshift	Unterstützung für die Konvertierung von Einschränkungen hinzugefügt.
Vertica	Amazon Redshift	Es wurde ein Fehler behoben, bei dem ein Bewertungsbericht nicht als Datei mit kommagetrennten Werten (CSV) gespeichert wurde.

Versionshinweise für Build 659 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Der Assistent für neue Projekte, der einen kombinierten Bewertung sbericht für mehrere Quelldatenbanken generiert, wurde verbessert.
Alle	Alle	Es wurde ein Problem behoben, bei dem das Erweiterungspaket in Projekten, die mehrere Quell- und Zieldatenbanken enthalten, nicht erstellt wurde.
Alle	Alle	Verbesserte Konvertierung von SQL-Code, der in den Quellcode der Anwendung eingebettet ist.
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, Skripts aus verschiedenen Ordnern in der AWS SCT Befehlszeilenschnittstelle auszuführen.
Alle	Amazon Redshift	Die Warnmeldung, die angezeigt wird, wenn Benutzer in Migration sprojekten mit der virtuellen Zieldatenbankplattform Amazon Redshift die Option Optimierung ausführen wählen, wurde verbessert.
Alle	Aurora PostgreSQL	Unterstützung der PostgreSQL-Hauptversion 13 auf der Aurora PostgreSQL-Compatible Edition als Migrationsziel implementiert.
Alle	Amazon RDS für MySQL	Standardmäßig wurde die Codekonvertierung ohne Berücksichtigung der Groß- und Kleinschreibung implementiert.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Verbindung zu einer Quelldatenbank in der Befehlszeilenschnittstelle fehlschlug.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Die Konvertierung von Prozeduren, die `UPDATE` Anweisungen mit Join-Bedingungen enthalten, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Verbesserte Konvertierung von Triggern, gespeicherten Prozeduren und Funktionen, die den Wert nach dem Gleichheitszeichen enthalten .
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Ein Transformationsfehler für Prozeduren mit der `DELETE` Anweisung und dem `OR` Operator wurde behoben.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Die Konvertierung der `OUTPUT` Klausel wurde verbessert.
Microsoft SQL Server DW	Amazon Redshift und AWS Glue	Verbesserte Konvertierung des `NUMERIC` Datentyps.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung von Ansichten, die einen Tabellenalias mit demselben Namen wie die Originaltabelle haben.
Microsoft SSIS	AWS Glue	Es wurde ein Problem behoben, bei dem die AWS Glue Verbindun gsinformationen nicht im Fenster Verbindungen konfigurieren angezeigt wurden.
Netezza	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, die Ausführung von CDC-Datenmigrationsaufgaben (Change Data Capture) täglich zu wiederholen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Registerkarte „Aufgaben“ inaktiv wurde, nachdem die Registrierung eines Datenextraktionsagenten aufgehoben wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Bestätigung der Registrierung des Datenmigrationsagenten nicht auf der Benutzeroberfläche angezeigt wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Verbindung zu einer Quelldatenbank mit einem Loader-Fehler fehlschlug.
Netezza	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Datenmigrationsagenten nach dem Öffnen eines gespeicherten Projekts nicht ausgeführt werden konnten.
Oracle	Amazon RDS für Oracle	Unterstützung von Oracle Unified Auditing implementiert.
Oracle	PostgreSQL Aurora PostgreSQL	Implementierung der SQL-Code-Konvertierung in C#-Anwendungen. Weitere Informationen finden Sie unter SQL-Code in C#-Anwendungen.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde eine neue Transformationslogik für Objektnamen implementiert, bei denen die Groß- und Kleinschreibung beachtet wird, um die Sichtbarkeit von Änderungen bei der Codekonvertierung zu verbessern. AWS SCT konvertiert Objektnamen in Großbuchstaben in Kleinbuchstaben. Das Gegenteil ist auch der Fall; AWS SCT konvertiert Objektnamen in Kleinbuchstaben in Großbuchstaben. Andere Objektnamen und reservierte Wörter werden unverändert konvertiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Verbesserte Konvertierung von Hash-Partitionen ohne `NOT NULL` Einschränkung.
Oracle	Aurora PostgreSQL	Unterstützung für die Konvertierung von Oracle `CHECK`, `FOREIGN KEY`, und `NOT NULL` Constraints mit der `ENABLE NOVALIDATE` Klausel hinzugefügt.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die falschen Werte für Fließkommazahlen migriert wurden.
Oracle DW	Amazon Redshift und AWS Glue	Es wurde ein Problem mit leeren Spalten im Bewertungsbericht zur Datenbankmigration in einer Datei mit kommagetrennten Werten (CSV) behoben.
SAP ASE	PostgreSQL Aurora PostgreSQL	Ein Problem mit einer unerwarteten Unterbrechung der Konvertierung wurde behoben.
Snowflake	Amazon Redshift	Die Konvertierung des `VARIANT` Datentyps wurde verbessert.
Teradata	Amazon Redshift	Die Konvertierung der `COLLECT STATISTICS` Anweisung wurde verbessert.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 nicht angezeigt wurde, wenn verschachtelte Ansichten mit `PERIOD` Spalten konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift und AWS Glue	Es wurde ein Problem behoben, bei dem eine virtuelle AWS Glue Zielplattform nach dem Öffnen eines gespeicherten Projekts nicht in der Benutzeroberfläche angezeigt wurde.
Teradata BTEQ	AWS Glue	Es wurde ein Problem behoben, bei dem die Konvertierung auf eine virtuelle AWS Glue Zielplattform nach dem Öffnen eines gespeiche rten Projekts nicht unterstützt wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Die Syntaxhervorhebung des konvertierten Codes wurde verbessert.
Teradata BTEQ	Amazon-Redshift-RSQL	Die Überprüfung der Parameterwerte nach dem Upload wurde implementiert. Nicht unterstützte Werte werden auf der Registerkarte Variablen hervorgehoben.
Vertica	Amazon Redshift	Die Konvertierung von Aggregatfunktionen wurde implementiert.
Vertica	Amazon Redshift	Die Konvertierung von Projektionen in materialisierte Ansichten wurde implementiert und die Benutzeroberfläche verbessert, die den Quellcode von Projektionen anzeigt.

Versionshinweise für AWS SCT Build 658

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Integration mit bereitgestellt AWS Secrets Manager. Sie können jetzt die Anmeldeinformationen für die Datenbankverbindung verwenden, die in Secrets Manager gespeichert sind.
Alle	Alle	Unterstützung für Skripts im YAML-Format in der AWS SCT Befehlszeilenschnittstelle hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Unterstützung von Amazon S3 S3-Schnittstellenendpunkten (VPCE) in Datenextraktionsagenten implementiert.
Alle	Amazon Redshift	Unterstützung für die virtuelle Zieldatenbankplattform Amazon Redshift wurde zusätzlich zu dem bereits unterstützten Amazon Redshift und AWS Glue dessen Kombination hinzugefügt.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem mit der Option Als SQL speichern der konvertierte SQL-Code nicht in einer Datei gespeichert wurde.
IBM Db2 (LUW) Microsoft Azure SQL Database Microsoft SQL Server Oracle SAP ASE	Aurora MySQL	Verbesserte Konvertierung zur Unterstützung neuer Funktionen der Amazon Aurora MySQL-Compatible Edition mit MySQL 8.0-Kompatibilität.
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 810 bei Bedarf nicht angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Prozeduren mit `UPDATE`, `DELETE`, und `INSERT` -Anweisungen wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 7810 bei Bedarf nicht angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung einer `EXEC` Anweisung, die in einer Anweisung verschachtelt ist, wurde verbessert. `IF...ELSE`
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von indizierten Ansichten wurde verbessert.
Netezza	Amazon Redshift	Verbesserte Datenmigrationsagenten durch die Nachverfolgung von Live-Transaktionen bei Volllast im CDC-Vorgang (Change Data Capture). Sie können Datenmigrationsaufgaben jetzt beenden, wenn der Start der CDC-Sitzung zu einem bestimmten Zeitpunkt geplant ist. Außerdem können Sie die Stufe der Fehlerprotokollierung in der Konsole sehen, nachdem Sie eine Aufgabe mit CDC beendet haben.
Oracle	Alle	Der Tabellenlader wurde verbessert, um sicherzustellen, dass Objekte mit Freigabeoptionen AWS SCT geladen werden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung der `SYSDATE` Funktion wurde verbessert und die Möglichkeit hinzugefügt, die Zeitzone in den Konvertierungseinstellungen zu ändern.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem dynamische Anweisungen nicht konvertiert wurden.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem der konvertierte Code keine vom System generierten Namen enthielt.
Oracle Oracle DW	Aurora PostgreSQL PostgreSQL	Die Konvertierung von `SELECT` Anweisungen, die innerhalb von Triggern verschachtelt sind, wurde verbessert.
Oracle DW	Amazon Redshift	Verbesserte Konvertierung der `TO_TIMESTAMP_TZ` Funktionen `TO_DATE` `TO_TIMESTAMP` , und im Erweiterungspaket.
Snowflake	Amazon Redshift	Es wurde eine Option hinzugefügt, um konvertierten SQL-Code in verschiedenen Dateien für jedes Objekt oder für jede Anweisung zu speichern.
Teradata	Amazon Redshift	Verbesserte Konvertierung der `CONCAT` Funktion.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung einer `SELECT` Anweisung, die in einer `WHERE` Klausel verschachtelt ist.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei `SET` dem `MULTISET` Tabellen falsch konvertiert wurden, nachdem Benutzer eine Tabelle gelöscht und neu erstellt hatten.
Teradata	Amazon Redshift	Die Konvertierung der Prozeduren, die eine `WITH` Klausel enthalten, wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung des `DATE` Datentyps.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem während der FastExport Skriptkonvertierung ein unerwarteter Transformatorfehler auftrat.
Teradata BTEQ	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung eines Join-Index in eine materialisierte Ansicht hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung einer `TITLE` Definition, die mehrere Zeilen umfasst, wurde hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Größe eines Geodatentyps nicht konvertiert wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Parameternamen in Kleinbuchstaben umgewandelt wurden.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die in einer `MACRO` Anweisung verschachtelt ist, nicht konvertiert wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Vertica	Amazon Redshift	Die Konvertierung des `ALL` Operators wurde verbessert.
Vertica	Amazon Redshift	Es wurde ein Problem behoben, bei dem die `Use Union all view?` Option in den Konvertierungseinstellungen nicht angewendet wurde.
Vertica	Amazon Redshift	Verbesserte Konvertierung der `TIME WITH TIMEZONE` Datentypen `TIME` und.
Vertica	Amazon Redshift	Ein Problem beim Laden von Flex-Tabellen wurde behoben.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 657

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Apache Log4j wurde auf Version 2.17 aktualisiert, um Sicherhei tslücken zu beheben.
Alle	Amazon Redshift	Verbesserte Projekte zur Schemaoptimierung, bei denen wichtige Verwaltungsstatistiken nicht im Projekt gespeichert wurden. AWS SCT
Amazon Redshift	Amazon Redshift	Ein Problem mit der Aktualisierung der Serverinformationen wurde behoben.
Apache Cassandra	Amazon-DynamoDB	Ein Problem mit Zuordnungsregeln bei der Verwendung der AWS SCT Befehlszeilenschnittstelle wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Apache Cassandra	Amazon-DynamoDB	Es wurde ein Problem behoben, bei dem die Migrationsaufgabe aufgrund eines aktualisierten Titels im Zertifikat nicht erstellt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, sodass das Aktionselement 7672 bei der Konvertierung von Microsoft SQL Server-Prozeduren mit dynamischem SQL nicht angezeigt wurde.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Tabellenwertfunktionen.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das `OUT` Argument in einer gespeicherten Prozedur mit dem Standardrückgabewert nicht in das `INOUT` Argument konvertiert wurde.
Greenplum	Amazon Redshift	Verbesserte Optimierungsstrategien, indem die am häufigsten verwendeten Tabellen und Spalten in der `QueryLog` Tabelle gefunden wurden.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Probleme bei der Konvertierung der folgenden Dateien wurden behoben: • Zuweisungsoperator für Zeichenkettenverkettung () += • `SCOPE_IDENTITY` Funktion • `varchar(max)` -Datentyp

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Ansichten mit nicht unterstützten Funktionen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem nicht unterstützte Funktionen als Argument für eine andere Funktion falsch konvertiert wurden.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Konvertierung von Verweisen auf Übergangstabellen wurde verbessert.
Microsoft SQL Server DW	Amazon Redshift	Die Kategorie Aggregatfunktionen wurde dem Metadatenbaum der Quelldatenbank hinzugefügt.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung des `TIME` Datentyps.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem `DROP CREATE` Skripts nicht gespeichert wurden, wenn eine virtuelle Zieldatenbankplattform verwendet wurde.
Microsoft SQL Server-Integrationsdienste	AWS Glue	Es wurde ein Problem behoben, bei dem die Skripts von Quellobjekten nicht in der Benutzeroberfläche angezeigt wurden.
Netezza	Amazon Redshift	Verbesserte Optimierungsstrategien durch Auswahl der Faktentabelle und geeigneter Dimensionen für die Kollokation.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem bei der korrekten Konvertierung von Oracle-Triggern behoben, die Sequenznummern verwenden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Ansichten mit öffentlichen Datenbank-Links wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Verbesserte Optimierungsstrategien durch Analyse der Kardinalität von Indexspalten.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem benutzerdefinierte benutzerdefinierte Skalarfunktionen mit Zeichenkettenverkettung falsch konvertiert wurden.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Option Als SQL speichern nicht in der Benutzeroberfläche angezeigt wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Erfassung von Statistik en mit der `LOADER ERROR` Ausnahme fehlschlug.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Option Bericht erstellen nicht in der Benutzeroberfläche angezeigt wurde.
Teradata	Amazon Redshift	Die Konvertierung der `CAST` Funktion wurde verbessert.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung für wurde behoben`ST_Line_I nterpolate_Point` .
Teradata	Amazon Redshift	Ein unerwarteter Wert wurde aus dem Python-Bibliothekspfad entfernt.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Resolver-Fehler behoben, der bei der Konvertierung mehrerer FastLoad Skripte auftrat.
Teradata BTEQ	Amazon-Redshift-RSQL	Verbesserte Konvertierung der `DATABASE` Befehls- und Geometrie datentypen.
Teradata BTEQ	AWS Glue	Ein Problem mit einer falschen Synchronisation der Quell- und Zielskripts in der Benutzeroberfläche wurde behoben.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 656

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für mehrere Quell- und Zieldatenbanken innerhalb eines Projekts hinzugefügt. Benutzer können jetzt Zuordnungsregeln erstellen, die unterschiedlichen Datenbankschemas und Zielplatt formen im selben Projekt entsprechen.
Alle	Alle	Unterstützung für virtuelle Zieldatenbankplattformen hinzugefügt. Benutzer müssen jetzt keine Verbindung zu einer Zieldatenbank herstellen, um zu sehen, wie ihr Quelldatenbankschema AWS SCT konvertiert wird.
Alle	Alle	Verbesserungen der Benutzeroberfläche: • Die Optionen Connect zum Server herstellen und Verbindung zum Server trennen wurden den Quell- und Ziel-Metadatenbäumen hinzugefügt. • Es wurde eine Option hinzugefügt, um einen Datenbankserver aus dem AWS SCT Projekt zu entfernen.
Cassandra	Amazon-DynamoDB	Es wurde ein Suchproblem behoben, bei dem die `CASSANDRA_HOME` Variable keinen Schrägstrich (`/`) hinter `cassandra.yaml` oder dem `conf` Ordner enthielt.
Cassandra	Amazon-DynamoDB	Unterstützung für Amazon Machine Image (AMI) für Amazon Linux 2 hinzugefügt.
Cassandra	Amazon-DynamoDB	Verbesserte Fehlermeldung, die angezeigt wird, wenn ein falscher Schlüssel für Cassandra eingegeben wird.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Cassandra	Amazon-DynamoDB	Verbesserte Konvertierung durch Änderung einer Eigenschaft in der `cassandra-env.yaml` Datei in Abhängigkeit von der Version der Zieldatenbank.
Cassandra	Amazon-DynamoDB	Die Java-Version auf dem Ziel-Cassandra Datacenter wurde auf 1.8.0 erhöht.
Greenplum	Amazon Redshift	Verbesserte Optimierungsstrategien in den Projekteinstellungen.
Greenplum	Amazon Redshift	Es wurde ein Datenmigrationsproblem behoben, bei dem Objekte mit diesem Fehler nicht auf die Datenbank angewendet wurden:`An I/O error occurred while sending to the backend .`
Greenplum Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die `Apply RTRIM to string columns` Option nicht in der Benutzeroberfläche angezeigt wurde.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Unterstützung für Babelfish for Aurora PostgreSQL als Zielplatt form hinzugefügt. Benutzer können jetzt einen Bewertungsbericht erstellen, um die Migration von SQL Server zu Babelfish for Aurora PostgreSQL abzuschätzen.
Netezza	Amazon Redshift	Verbesserte Optimierungsstrategien in den Projekteinstellungen.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit implementiert, eindeutige Namen für Indizes zu generieren.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	Aurora PostgreSQL PostgreSQL	Ein Problem mit einer doppelten Indexspalte im Zielskript wurde behoben.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Optionen Leere Schemas ausblenden, Leere Datenbanken ausblenden und Systemdatenbanken/Schemas ausblenden nicht in der Benutzero berfläche angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von MultiLoad Teradata-Jobskripten in Amazon Redshift RSQL-Skripts hinzugefügt.
Teradata	Amazon-Redshift-RSQL	Ein Problem mit der Konvertierung von Substitutionsvariablen in und-Skripten wurde behoben. FastLoad FastExport
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Aktionspunkte nach dem Wechsel von der Registerkarte „Zusammenfassung“ nicht auf der Registerkarte „Aktionspunkte“ angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem nach dem Generieren eines Berichts während der FastExport Skriptkonvertierung ein Fehler auftrat.
Teradata	Amazon-Redshift-RSQL	Formatierungsprobleme nach der Konvertierung von Shell-Skripten wurden behoben.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, sodass AI 13177 jetzt im konvertie rten Skript kommentiert wird.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung von temporalen Tabellen wurde behoben.
Teradata	Amazon Redshift	Die Konvertierung der `SET QUERY_BAND` Anweisung wurde verbessert.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung der `NORMALIZE` Operation wurde behoben.
Vertica	Amazon Redshift	Die Beschreibung von AI 17008 wurde verbessert.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 655

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, durch das sichergestellt wurde, dass alle Bewertungsprobleme in Berichten angezeigt werden, wenn FastLoad oder verwendet MultiLoad wird.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von FastExport Teradata-Jobskripten in Amazon Redshift RSQL-Skripts hinzugefügt.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, das sicherstellte, dass die Aktion „Manifest in S3 speichern“ im Offline-Modus aktiviert ist, wenn sie verwendet wird. FastLoad
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, um sicherzustellen, dass Zuordnung sregeln für Skripte wie angewendet werden FastLoad.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Die unterstützte Mindesttreiberversion für Greenplum wurde auf 42.2.5 erhöht.
Greenplum	Amazon Redshift	Es wurde eine Verbindung zu Greenplum über SSL mit der Treiberversion 42.2.5 oder höher hinzugefügt.
Oracle DW	Amazon Redshift	Verbesserte Unterstützung für die Ausführung benutzerdefinierter benutzerdefinierter Skalarfunktionen (UDF) innerhalb einer anderen UDF.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem Funktionen mit diesem Fehler nicht auf die Datenbank angewendet wurden:`Failed to compile udf` .
Oracle DW	Amazon Redshift	Verbesserte Konvertierung durch Verwendung geeigneter Typdeklarationen, z. B. `pls-type` für %ROWTYPE-Parameter.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Probleme mit der Bewertung des Informationstyps nicht im Bericht angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Transformer-Fehler nach der Konvertierung einiger Skripts behoben.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, sodass ein Problem jetzt im konvertierten Skript kommentiert wird.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem FastExport ->EXPORT -> 'null' statt 'CAST' nach der Konvertierung angezeigt wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem einige Funktionen eines Erweiterungspakets fehlschlugen, wenn die Treiberversion `Cause: [JDBC Driver]String index out of range: 0` 1.2.43 verwendet wurde

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	SET-Tabellenkonvertierung — SET-Tabellenemulation für Insert-Select-Anweisungen hinzugefügt.
Teradata	Amazon Redshift	CAST — Unterstützt zusätzliches Casting von Datentypen.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung für „other_current_time_01" wurde behoben
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: Verbesserte Konvertierung von Teradata-Befehlen — Field FastExport
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: Verbesserte Konvertierung von Teradata-Befehlen — Layout FastExport
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem sich das Zielskript von Objekten mit SAVE EXCEPTIONS STATEMENT nach der erneuten Konvertierung änderte.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem nach der `proc_cursor_with_calc_columns` Konvertierung ein falsches Feld in der ORDER BY-Klausel angegeben wurde.
Oracle	PostgreSQL Aurora PostgreSQL	Behoben: Bei einer ASSOCIATIVE COLLECTION-Konvertierung ist eine zusätzliche `aws_oracle_ext$array_id$temporary` Variablendeklaration erforderlich.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Behoben: Die falsche Konvertierung eines PRIMÄRSCHLÜSSELS mit demselben Namen eines INDEX, der derselben Tabelle gehört.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 654

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem mit dem Fehler beim Parsen von Pseudosplten für hierarchische Abfragen und PRIOR-Spalten behoben.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem ein mehrzeiliger Kommentar mit einem Schrägstrich und einem Sternchen (/*) korrekt konvertiert wurde.
Oracle	PostgreSQL Aurora PostgreSQL	Dem Erweiterungspaket wurde die Systemansicht-Emulation USER_COL_COMMENTS hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Verbesserte Konvertierung von Literalen in Anführungszeichen.
DB2 LUW	PostgreSQL Aurora PostgreSQL	Verbesserte Konvertierung von LABEL-Anweisungen, die Labels in den Beschreibungen von Tabellen, Ansichten, Aliasnamen oder Spalten hinzufügen oder ersetzen.
Oracle	Keine	Die Systemtabelle SYS.USER$ wurde durch die Ansicht DBA_USERS ersetzt und die Abfragen wurden verbessert.
Oracle DW	Amazon Redshift	Oracle DW-Metadatenabfragen wurden aktualisiert.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von Shell-, Teradata- und Teradata FastLoad Basic Teradata Query (BTEQ) -Skripts in Amazon Redshift RSQL-Skripts hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem „merge_01" falsch konvertiert wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Das Problem, dass End or Identify (EOI) am Ende eines Skripts in einer neuen Zeile angezeigt wird, wurde behoben.
Azure Synapse	Amazon Redshift	Die Fehlermeldung, die angezeigt wird, wenn ein falsches Passwort für Azure Synapse eingegeben wurde, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Konvertierung von UPDATE-Anweisungen wurde verbesser t, sodass der richtige Aliasname gemäß dem Teradata-Standard übernommen wird.
Teradata	Amazon Redshift	Es wurde ein Fehler bei der Cursorkonvertierung behoben, bei dem keine Aktionen empfangen wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem bei einer TD_NORMAL IZE_OVERLAP-Konvertierung Zeilen gelöscht wurden.
Teradata	Amazon Redshift	Unterstützt jetzt die strikte Datumsprüfung für die erweiterte TO_DATE-Funktion.
Teradata	Amazon Redshift	Verbesserte Konvertierung der integrierten Funktion TO_NUMBER (n).
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Kategorie Schemas in der Metadatenstruktur fehlte.
Greenplum	Amazon Redshift	Die Auswahl GP_SEGMENT_ID wurde zur Liste hinzugefügt, wenn eine virtuelle Partition für eine Greenplum-Tabelle erstellt wurde.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem Funktionen nicht auf das Ziel angewendet wurden.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem nach der Konvertierung ohne AI 9996 ein Transformationsfehler auftrat.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem beim Öffnen des Erweiteru ngspaket-Assistenten ein Fehler protokolliert wurde.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem ein falscher Kommentarstil für Redshift-Python-Funktionen verwendet wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem ein Netezza-Redshift-E rweiterungspaket mit einem Profil nicht erstellt werden konnte. AWS

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Die Konvertierung des Befehls SESSIONS wurde verbessert. FastLoad
Teradata	Amazon-Redshift-RSQL	Verbesserte Bewertungsberichte für FastLoad Skripts.
Teradata	Amazon-Redshift-RSQL	FastLoad Die WRITER-Aktion „In S3 speichern“ wurde implementiert.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Schaltflächen Skript FastLoad speichern\ Manifest in S3 speichern nicht aktiv waren.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem FastLoad multifile_script nach der Konvertierung nur eine Manifestdatei anstelle der erwartete n drei Dateien erstellte.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem FastLoad zusätzliche Ordner in einem S3-Pfad angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem der falsche Name der Manifestdatei in einem S3-Pfad angegeben FastLoad war.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 653

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde die Fähigkeit implementiert, dynamisches SQL zu konvertieren, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde.
Oracle	PostgreSQL Aurora PostgreSQL	Verbesserte dynamische SQL-Konvertierung: In-Parameter als Bind-Variablen.
Oracle DW 18, 19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Redshift implementiert: verbesserte integrierte Konvertierungsfunktionen. Aggregiertes LISTAGG; Analytisches LISTAGG.
Oracle DW 18,19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Redshift implementiert: Neue Funktionen abfragen.
Vertica	Amazon Redshift	Verbesserungen bei der Konvertierung von Vertica nach Redshift implementiert: SSL-zu-JDBC-Verbindung mit SSL=True.
MS SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von MS SQL Server nach Redshift: Externe Tabellen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Interval-Datentypen, arithmetische Operationen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Support für laterale Spaltenaliase.
Oracle	Keine	Die folgenden Loader-Abfragen verwenden jetzt statt: `DBA_USERS` `SYS.USER$`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		• get-tree-path-list- by-name-path .sql • estimate-table-or-view- .sql constraints-by-schema • estimate-table-or-view- .sql constraints-by-selected-schemas
Teradata	Amazon Redshift	Verbesserte Ausrichtung von Kommentaren, wenn SCT Teradata-Makros in gespeicherte Redshift-Prozeduren konvertiert.
Oracle DW	Amazon Redshift	Verbesserte Konvertierung von Date/Timestamp-Formatelemen ten:`TO_DATE`, und `TO_TIMESTAMP TO_TIMESTAMP_TZ`
Teradata	Amazon Redshift	Der Fehler bei der Teradata-Cursorkonvertierung wurde behoben.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, das dazu führte, dass Attribute von `TD_NORMALIZE_OVERLAP` während der Konvertierung gelöscht wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die `MAX` Funktion ignoriert wurde, als SCT eine Abfrage konvertierte.
Teradata	Amazon Redshift	SCT konvertiert jetzt die Teradata CHARACTERS-Funktion in die Redshift LENGTH-Funktion.
Teradata	Amazon Redshift	SCT unterstützt jetzt die Konvertierung von FORMAT in TO_CHAR für die am häufigsten verwendeten Formate.
Alle	Alle	Verbesserte Konvertierung verschlüsselter Routinen.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 652

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	PostgreSQL	App-Sperren für `sp_getapplock` und `sp_releaseapplock` Funktionen hinzugefügt.
Keine	Amazon Redshift	Verbesserung der Befehlszeilenschnittstelle (CLI): Implementierter Skriptbefehlsmodus.
Oracle	PostgreSQL Aurora PostgreSQL	Das Sampling von Routineparametern wurde in dynamischem SQL implementiert.
Oracle	PostgreSQL Aurora PostgreSQL	Verbesserungen bei der Konvertierung von dynamischem SQL, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL	Jede Lambda-Funktion wird nur einmal über eine Richtlinie bereitgestellt und konfiguriert, und gängige Lambda-Funktionen werden für alle möglichen Quellen wiederverwendet.
DB2 LUW	PostgreSQL	Es wurde ein Problem behoben, das zu der Fehlermeldung „9996 — Schweregrad kritisch — Transformer-Fehler ist aufgetreten“ führte, wenn DB2 LUW als Quelle verwendet wurde.
Teradata	Amazon Redshift	Support für rekursive Tabellenausdrücke bei der bevorstehenden Markteinführung von Amazon Redshift.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse	Amazon Redshift	Regeln zur Schemaoptimierung implementiert.
Teradata	Amazon Redshift	Support Zeitzonenkonvertierung von Teradata-Makros in gespeicherte Redshift-Prozeduren.
Teradata	Amazon Redshift	Support Arithmetik für PERIOD-Werte.
Teradata	Amazon Redshift	Support die Konvertierung von Teradata Recursive Common Table Expressions (RECURSIVE CTE).
Teradata	Amazon Redshift	Support Kennungen, bei denen Groß- und Kleinschreibung beachtet wird, über die Benutzereinstellung,`enable_case_sensitive_identifier` . „COLUMN_NAME“ und „Column_Name“ werden also zu unterschiedlichen Spaltennamen.
Teradata	Amazon Redshift	Das Problem mit dem Datentyp Dezimal wurde behoben, sodass Dezimalfelder mit derselben Genauigkeit konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem mit der arithmetischen Intervallkonvertierung behoben, sodass die Intervallarithmetiksubtraktion korrekt konvertiert wurde.
Teradata	Amazon Redshift	Die Umwandlung vom Typ Teradata NUMBER in DATE wurde verbessert.
Teradata	Amazon Redshift	Die Umwandlung von Teradata vom Typ DATE in NUMBER wurde verbessert
Teradata BTEQ	Amazon Redshift	Verbesserte Konvertierung des PERID-Datentyps.
Teradata	Amazon Redshift	Es wurde ein Problem beim Laden von Metadaten für eine Tabelle mit GEOMETRY-Spalten behoben, sodass sie jetzt korrekt aus Teradata geladen wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Support die Konvertierung von Merge-Anweisungen bei der Konvertierung von Teradata-Makros in gespeicherte Redshift-Prozeduren.
Teradata	Amazon Redshift	Verbesserte Konvertierung einfacher Makros bei der Migration von Teradata zu Redshift.
Teradata	Amazon Redshift	Es wurde sichergestellt, dass bei der Konvertierung der Teradata UPDATE-Anweisungen der richtige Aliasname gemäß dem Teradata-Standard übernommen wurde.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 651

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Verbesserte AWS SCT Berichte zur Aktualisierung der Links zu den aufgeführten empfohlenen Konversionsaktionselementen.
MS SQL Server	PostgreSQL	Unterstützung für die Konvertierung von `STR()` Funktionen hinzugefügt.
MS SQL Server	PostgreSQL	Unterstützung für die Konvertierung des bitweisen EXOR-Operators (^in Microsoft SQL Server) in PostgreSQL als Operator hinzugefügt. #
Oracle	PostgreSQL	Es wurde ein Problem behoben, bei dem die AWS SCT `aws_oracle_ext.UNISTR(null)` Erweiterungspack-Funktion `NULL` auf einem PostgreSQL-Ziel hängen blieb. AWS SCT behandelt jetzt die. `NULL`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata BTEQ	Amazon-Redshift-RSQL	Konvertierungsverbesserungen zur Behebung eines Problems, bei dem die Konvertierung von Amazon Redshift RSQL MERGE zu einem Transformationsfehler führte.
Oracle DW	Amazon Redshift	Verbesserte integrierte Funktionen implementiert.
Oracle DW	Amazon Redshift	Es wurden Verbesserungen hinzugefügt, die auf Funktionen für Metadaten basieren, darunter automatische Listenpartitionierung (TBL_PART_LIST_AUTO), Liste mit mehreren Spalten (TBL_PART_MULTI_LIST) und Interval-Referenz (TBL_PART_RANGE_INTVAL_REF).
Keine	Amazon Redshift	Die Beschränkungen der `UNION ALL` Partitionstabellen für physische Partitionen, die für Konvertierungen verwendet werden, wurden erhöht.
Teradata	Amazon Redshift	Im Rahmen der Bewertungsberichte wurden Verbesserungen bei der Konvertierung vorgenommen.
Teradata	Amazon Redshift	Konvertierungsverbesserungen bei komplexen Teradata MACRO-Konvertierungen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von Teradata-Makros in gespeicherte Amazon Redshift Redshift-Prozeduren beim Auskommentieren von nicht unterstütztem SQL.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Konvertierung von Teradata-Makros in gespeicherte Amazon Redshift Redshift-Prozeduren zu falschen Aliasnamenreferenzen führte.
Teradata	Amazon Redshift	Die Konvertierung der Teradata-Anweisung wurde verbessert. `QUALIFY`
Teradata	Amazon Redshift	Die Konvertierung wurde verbessert, um Kommentare an Amazon Redshift zu übertragen und einen Verlauf der an der Ansicht vorgenommenen Änderungen beizubehalten.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die RESET WHEN-Klausel nicht zur korrekten Konvertierung führte.
Teradata BTEQ	Amazon Redshift	Verbesserte Konvertierung von BTEQ-Skripten, die MERGE-Anw eisungen enthalten.
Teradata	Amazon Redshift	Es wurden integrierte Funktionen hinzugefügt, um die Konvertierung von Feldern vom Datentyp PERIOD zu verbessern.
Microsoft SQL Server	Amazon Redshift	Verbesserte Zuordnung von Transformationsdatentypen für den TIME-Datentyp.
Alle	Alle	Zugriff auf die Erstveröffentlichung des AWS Schema Conversion Tool CLI-Referenzhandbuchs im PDF-Format hinzugefügt. Siehe AWS Schema Conversion Tool CLI-Referenz.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 650

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Aktualisierte und erweiterte Verwendung von Extraktor-Agenten, einschließlich: • Eine Konfiguration für die Verwendung mit gemeinsam genutztem Speicher und einem dedizierten Kopieragenten. • Exportieren und Importieren von Datenextraktionsaufgaben von einem Projekt in ein anderes. •

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		Support für Azure SQL Data Warehouse (Azure Synapse) als Quelle. • Verwendung der nativen Netezza-Partitionierung. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.
Alle	Amazon RDS PostgreSQL 13	AWS SCT unterstützt jetzt Amazon RDS PostgreSQL 13 als Ziel.
Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung einer Ergebnismenge von einer Microsoft SQL Server-Prozedur in ein Aurora PostgreSQL-Ziel wurde verbessert.
Oracle DW	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert.
Oracle DW	Amazon Redshift	Verbesserungen bei der Konvertierung dynamischer SQL-Anweisungen wurden implementiert.
Oracle DW	Amazon Redshift	Verbesserungen an der SQL UDF-Konvertierung implementiert.
Oracle DW	Amazon Redshift	Die Meldung, die die Konvertierung von EXTERNEN TABELLEN AWS SCT nicht unterstützt, wurde geklärt.
Oracle DW	Amazon Redshift	Verbesserte integrierte Konvertierungsfunktionen.
Teradata BTEQ	Amazon-Redshift-RSQL	Verbesserte Handhabung von Substitutionsparametern in BTEQ-Skripten bei Verwendung der GUI. AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW Microsoft SQL Server Azurblau Azure Synapse	Alle	Die mindestens unterstützte JDBC-Treiberversion für Microsoft SQL Server, Azure, Azure Synapse wurde aktualisiert.

Behobene Probleme:

- Teradata: Zusätzliche Verbesserungen bei der Makrokonvertierung [BEHOBEN]
- Sonderzeichen fehlten im Ziel, was zu SQL-Fehlern führte. Sie mussten überarbeitet werden, um sie wieder einzufügen [BEHOBEN]
- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 649

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von MSSQL zu Amazon Redshift zur Unterstützung temporaler Tabellen.
Oracle DW	Amazon Redshift	Integrierte Funktionserweiterungen wurden implementiert, wie z. B.: Konvertierungs-Funktionen • TO_BINARY_DOUBLE •

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		TO_BINARY_FLOAT • TO_NUMBER • TO_DATE • TO_TIMESTAMP • TO_TIMESTAMP_TZ • TO_DSINTERVAL • TO_YMINTERVAL • VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	Funktionserweiterungen für Approximate Query Processing wurden implementiert, wie z. B.: Aggregationsfunktionen • ANY_VALUE • APPROX_COUNT_DISTINCT • APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • ZU_APPROX_COUNT_DISTINCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Konvertierungsverbesserungen für die auto Sortierung und die Auswahl der Verteilungsschlüssel von Teradata wurden implement iert. Die DB-Engine wählt automatisch Verteilungs- und Sortiersc hlüssel aus. Es wurde ein Optionsfeld mit der Bezeichnung Automatis che Tabellenoptimierung von Amazon Redshift verwenden, um Einstellungen für aktuelle Projekte > Optimierungsstrategien > Anfängliche Strategie für die Schlüsselauswahl zu aktivieren, eingeführt.
Teradata	Amazon Redshift	Der AWS SCT Tabellenlader wurde verbessert, um sicherzustellen, dass alle Tabellen aus Teradata AWS SCT geladen werden.
Teradata	Amazon Redshift	Konvertierungsverbesserungen wurden implementiert, sodass Amazon Redshift korrelierte Unterabfragemuster unterstützt, die eine einfache WHERE NOT EXISTS-Klausel enthalten.
Teradata	Amazon Redshift	Unterstützung für die Verwendung von ECHO-Befehlen in Makros hinzugefügt.
DB2 LUW	PostgreSQL Aurora PostgreSQL	Unterstützung für die Konvertierung von DYNAMIC RESULTS SETS implementiert, einschließlich: • Cursorklausel WITH RETURN/WITH RETURN TO CLIENT • Konvertierung von DYNAMIC RESULT SETS-Routinek
Microsoft SQL Server Oracle DB2 LUW SAP ASE	Aurora PostgreSQL	Unterstützung für aktuelles Aurora RDS PostgreSQL als Ziel implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	Unterstützung für MariaDB 10.5 als Ziel implementiert.
Microsoft SQL Server	MariaDB	Implementierte Unterstützung von INSERT-RETURNING, die eine Ergebnismenge der eingefügten Zeilen zurückgibt.
Oracle	Aurora PostgreSQL	Unterstützung der XMLFOREST-Funktion für die Konvertierung von Oracle nach Aurora PostgreSQL hinzugefügt.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 648

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Amazon Aurora PostgreSQL-Compatible Edition	Benutzerdefinierter Anwendungsmodus für das Aurora PostgreSQL-Erweiterungspaket implementiert: Operatoren für numerische/ Datums- und Texttypen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL	Aurora PostgreSQL Lambda Invoke-Konfiguration implementiert: Erstellung der aws_lambda-Erweiterung; IAM-Rollenzuweisung an den Aurora PostgreSQL-Cluster. • Oracle — WebAgent E-Mails, Jobs, Warteschlangen, Dateien • DB2—E-Mails, Aufgaben, Dateien • Microsoft SQL Server — E-Mails, Agent
Oracle	PostgreSQL	Das Refactoring zur Konvertierung von FORALL-Anweisungen wurde implementiert: • FORALL-Anweisung • FÜR ALLE... AUSNAHMEN SPEICHERN • RÜCKKEHR MIT BULK COLLECT • SQL%BULK_EXCEPTIONS-Systemsammlung
Oracle DW 18, 19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert: verbesserte integrierte Konvertierungsfunktionen. Aggregiertes LISTAGG; Analytisches LISTAGG.
Oracle DW 18,19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert: Neue Funktionen abfragen.
Vertica	Amazon Redshift	Verbesserungen bei der Konvertierung von Vertica zu Amazon Redshift implementiert: SSL-zu-JDBC-Verbindung mit SSL=True.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von Microsoft SQL Server nach Redshift: Externe Tabellen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Interval-Datentypen, arithmetische Operationen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Support für laterale Spaltenaliase.

Behobene Probleme:

- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 647

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Microsoft SQL Server	RDS unterstützt jetzt die Datenbank-Mail-Funktion.
Microsoft SQL Server	MySQL	Implementierung des maximalen Namens für jeden Bezeichnertyp — Die maximale Länge von Objektnamen (z. B. Tabellen, Einschrän kungen, Spalten) in SQL Server beträgt 128 Zeichen. Die maximale Länge von Objektnamen in MySQL beträgt 64 Zeichen. Um konvertie rte Objekte in die MySQL-Datenbank zu schreiben, müssen Sie ihre Namen kürzen. Um doppelte Namen nach dem Ausschneiden zu vermeiden, müssen Sie den neuen Namen eine „Prüfsumme“ des ursprünglichen Objektnamens hinzufügen. Schneiden Sie Namen, die länger als 64 Zeichen sind, wie folgt aus: `[first N chars]() + "" + [checksum]()`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		`[first N chars] = 64 - 1 - [length of checksum string]` Zum Beispiel: `example_of_a_test_schema_with_a_name_length_greater_than_64_characters ?? example_of_a_test_schema_with_a_name_length_greater_than_64_9703`
Oracle	MySQL/ Aurora MySQL	Das Laden und Konvertieren von Kommentaren zu Speicherobjekten wurde implementiert. Zum Beispiel die Verarbeitung von Kommentaren zu Tabellen und die Verarbeitung von Kommentaren zu Tabellen-/ View-Spalten.
Teradata	Amazon Redshift	Unterstützung für die TIME-Datentypkonvertierung hinzugefügt.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung — TD_NORMALIZE_OVERLAP implementiert.
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung — SELECT mit WITH-Klausel; SELECT ohne FROM
Alle	Alle	AWS SCT Data Migration Service Assessor (DMSA) — Diese neue Funktion ermöglicht es Ihnen, mehrere Server zu bewerten und einen zusammenfassenden Bericht zu erhalten, der die beste Zielrichtung für Ihre Umgebung aufzeigt.
Alle	Alle	AWS SCT Assistent — Der Zielvergleich zeigt jetzt Unterschiede zwischen Zielen in einer einzigen Tabellenansicht.
Alle	Alle	Tree Filter UI — Der neu gestaltete Metadatenfilter verarbeitet komplexere Filtermuster.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Bewertungsbericht — Der neu gestaltete Abschnitt mit der Warnung bietet eine bessere Beschreibung und ein klareres Verständnis eines Problems.

Behobene Probleme:

- Allgemeine Verbesserungen
- Datenextraktoren — Die Unteraufgabe ist mit ConcurrentModificationException [RESOLVED] fehlgeschlagen.
- Microsoft SQL Server zu MySQL — maximale Identifikationslänge [AUFGELÖST].

Versionshinweise für AWS SCT Build 646

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	Verbesserte Implementierung des Modells im TM-Format.
Oracle	PostgreSQL	Die Maskenimplementierung im SP-Format bietet grundlegende Unterstützung für das SP-Suffix, nur für die englische Sprache.
Oracle	PostgreSQL	Behandlung langer Oracle-Objektnamen — verarbeitet AWS SCT nun lange Oracle-Objektnamen entsprechend dem Attribut für die maximale Länge des Zielbezeichners.
	Amazon Redshift	Amazon Redshift Redshift-Kodierung AZ64 mit AWS SCT — Kompressionskodierung AZ64 für einige Datentypen hinzugefügt
Teradata	Amazon Redshift	Unterstützung für die Konvertierung impliziter Transaktionen hinzugefügt.
Teradata	Amazon Redshift	Unterstützung für die integrierten Geodatenfunktionen von Teradata hinzugefügt: Methoden `ST_LineString`

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Greenplum-Sequenzkonvertierung — Die nächsten Elemente wurden zu den Eigenschaften-Tabs hinzugefügt: Minimalwert, Maximalwert, Inkrement, Zyklus.
Greenplum	Amazon Redshift	Resolver — Die Auflösung des Datentyps „char“ wurde hinzugefügt.
Greenplum	Amazon Redshift	Länge der Zeichenkonvertierung — Die PL/pgSQL-Konvertierung für den Zeichentyp wurde aktualisiert.
Greenplum	Amazon Redshift	Es wurde ein Problem mit der Auswahl des Greenplum-Verteilungsschlüssels behoben, bei dem eine Tabelle zwar über einen VERTEILUNGSSCHLÜSSEL verfügte, die Tabelle aber nicht als AWS SCT ZUFÄLLIG VERTEILT erkannt und abgerufen werden konnte.
Teradata	Amazon Redshift	Teradata-Cursorunterstützung — Unterstützung für die Cursorkonvertierung hinzugefügt.
Teradata	Amazon Redshift	Identitätsspalten — Unterstützung für die Konvertierung von Identitätsspalten hinzugefügt.
Teradata	Amazon Redshift	INTERVAL-Datentypen — Unterstützung für die Konvertierung von INTERVAL-Datentypen wurde hinzugefügt.

Behobene Probleme:

- Allgemeine Verbesserungen
- Greenplum: Die Konvertierung konnte aufgrund des Fehlers im Protokoll [BEHOBEN] nicht ausgeführt werden.
- MSSQL — PostgreSQL: Transformer-Fehler bei der Konvertierung der LAG-Funktion [BEHOBEN].
- MSSQL — PostgreSQL: SCOPE_IDENTITY [GELÖST].
- AWS SCT hängt in DW-Projekten [BEHOBEN].
- Benötigen Sie eine Zuordnungsregel, um zusätzliches Leerzeichen im Spaltennamen in AWS SCT [RESOLVED] zu entfernen.

Versionshinweise für AWS SCT Build 645

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Stellen Sie eine Lösung zur Lösung von Teradata-Ansichten bereit, die nicht vollständig qualifiziert sind (Namen der Ansichten oder nicht vollständig qualifizierte Objekte in der Ansicht).
Teradata	Amazon Redshift	Unterstützung der ASCII-Funktion zur Berechnung von Knoten hinzugefügt.
Teradata	Amazon Redshift	Wenn AWS SCT Multibyte-Daten in einer Teradata erkannt werden, die als `CHAR` definiert sind`CHAR(N)`, werden sie in Amazon Redshift `VARCHAR(3*N)` in diese umgewandelt.
Teradata	Amazon Redshift	Stellen Sie eine Teradata-Konvertierung zwischen Daten `CAST` und Zahlen bereit. • `SELECT Cast('2020-07-17' AS BIGINT)` • `SELECT Cast(20200630 - 19000000 AS DATE)`
Teradata	Amazon Redshift	Support die Konvertierung von `PERIOD` Teradata-Datentypen in zwei Amazon Redshift `TIMESTAMP` Redshift-Spalten: • `PERIOD(TIMESTAMP)` • `PERIOD(TIMESTAMP WITH TIMEZONE)`
Teradata	Amazon Redshift	Support Konvertierung der `RANK` Teradata-Funktion mit `RESET WHEN` Klausel.
Teradata	Amazon Redshift	Verbesserte Unterstützung von CAST bei expliziten Datentypk onvertierungen und impliziten CASTs On-Ausdrücken.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Nicht unterstützte korrelierte Unterabfragemuster melden. Weitere Informationen finden Sie unter Korrelierte Unterabfragen im Amazon Redshift Database Developer Guide.
Keine	Amazon Redshift	Verbesserte Tabellen schränken die Unterstützung für RA3 Knotentypen ein.
Teradata	Amazon Redshift	Unterstützung für die Extraktion von Teradata-Geodaten hinzugefügt. Weitere Informationen finden Sie unter Abfragen räumlicher Daten in Amazon Redshift im Amazon Redshift Database Developer Guide.
Microsoft SQL Server	PostgreSQL	Die Option wurde hinzugefügt,. `convert_procedures_to_function`

Behobene Probleme:

- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 644

Die Änderungen für die AWS SCT Versionen 1.0.643 wurden in der Version AWS SCT 1.0.644 zusammengeführt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Mehrere Konvertierungsverbesserungen. • Verbesserte Konvertierungen `QUALIFY` mit dem Tabellenalias. • Verbesserte Konvertierungen mit dem `IN` Operator. • Verbesserte Konvertierung mit dem `LIKE` Operator. •

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		Verbesserte Konvertierungen mit Hervorhebung von Problemen im konvertierten Code. • Verbesserte Konvertierungen mit ungewöhnlicher Reihenfolge von `WHERE QUALIFY` Klauseln in SQL. • Es wurden Transformatorfehler behoben, die während der `JOIN()` Konvertierung der Prozedur `UPD_FT_SVC_TRANS_BH_CBH_IND` aufgetreten sind. • Die Konvertierung von Makros in gespeicherte Prozeduren wurde verbessert. Es wurden spezielle AWS SCT CLI-Befehle hinzugefügt, die die bereitgestellten sql/bteq-Skripte analysieren und einen Bericht über die Anzahl der im Quellcode gefundenen Syntaxstrukturen generieren können. • Anzahl der BTEQ-Befehle • Anzahl der HANDLER • Anzahl der CAST-Fälle • Anzahl der DML/DDL-Fälle • Anzahl der aktualisierbaren Ansichten DMLs Ein Aktionselement für den Bewertungsbericht wurde hinzugefügt: Teradata-Spalten mit benutzerdefinierten Datumsformaten werden in Amazon Redshift nicht unterstützt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL/Aurora PostgreSQL	Es wurde eine Funktion zum Speichern von Installationsskripten für Erweiterungspakete hinzugefügt. Schweregrad für AI 5334 geändert. Die Leistung bei der Verwendung eines Datensatzes als Paketvariable `IMPLEMENTATION` wurde verbessert. Unterstützung für `XMLAGG` Aggregationsfunktionen hinzugefügt
IBM Db2	PostgreSQL/Aurora PostgreSQL	Das Laden und Konvertieren von Kommentaren zur Implementierung von Speicherobjekten wurde hinzugefügt.
MS SQL DW	Amazon Redshift	Verbesserung der Konvertierung: Problem mit behoben`PATINDEX`. Verbesserungen der Benutzeroberfläche: • Für die Implementierung des Quellbaums als SQL speichern. • Zusätzliche Logik zur Skriptgenerierung für mehrere Dateien hinzugefügt.
Vertica	Amazon Redshift	Verbesserung der Benutzeroberfläche: Als SQL für die Implementierung des Quellbaums speichern.

Behobene Probleme:

- Allgemeine Verbesserungen bei Konvertierungen zwischen Teradata und Amazon Redshift
- Allgemeine Fehlerbehebung und Verbesserungen der Benutzeroberfläche

Versionshinweise für AWS SCT Build 642

Änderungen für AWS Schema Conversion Tool Version 1.0.642.

> **Note**
>
> AWS Schema Conversion Tool (AWS SCT) Die Änderungen in Build 1.0.642 gelten für Windows, Ubuntu und Fedora. Es gibt keinen 1.0.642-Build für macOS.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SSIS	AWS Glue	Die Konvertierung von Microsoft SQL Server Integration Services (SSIS) ETL-Paketen in wurde implementiert. AWS Glue Weitere Informationen finden Sie unter SSIS wird in mit konvertiert AWS GlueAWS SCT.
Oracle	MariaDB/SQL MODE=CLE/MySQL/AmazonAurora MySQL	Der PL/SQL-Deklarationsabschnitt wurde in der WITH-Klausel implementiert.
Oracle	PostgreSQL/Aurora PostgreSQL	Unterstützung für `DBMS_SESSION.RESET_PACKAGE` und `DBMS_SESSION.MODIFY_PACKAGE` hinzugefügt.
Vertica	Amazon Redshift	Ermöglichen Sie den Export von SQL-Skripts aus einer Vertica-Datenbank nach Amazon Redshift.

Behobene Probleme:

- Verbesserung des Bewertungsberichts.
- Verbesserung der Benutzeroberfläche des Bewertungsberichts.
- Fügen Sie die Möglichkeit hinzu, die JVM-Einstellungen über die Benutzeroberfläche zu ändern.
- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 641

Änderungen für AWS Schema Conversion Tool Version 1.0.641.

> **Note**
>
> AWS Schema Conversion Tool (AWS SCT) Die Änderungen in Build 1.0.641 gelten für Windows, Ubuntu und Fedora. Es gibt keinen 1.0.641-Build für macOS.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle/ MS SQL/ MySQL/ PostgreSQL/Db2 LUW	Alle	Erstellen Sie Berechnungen des Zeitberichts in der CSV-Datei.
Teradata	Amazon Redshift	Unterstützung für die CSUM-Funktion hinzugefügt. Unterstützung für Geodatentypen von Teradata hinzugefügt.
Teradata	Alle	Unterstützung für die Konvertierung von IDENTITY-Spalten hinzugefügt.
Greenplum	Amazon Redshift	Unterstützung für den Verteilungsstil AUTO während der Greenplum-Tabellenkonvertierung hinzugefügt.
SAP ASE	Alle	Erstellen Sie Berechnungen des Zeitberichts in der CSV-Datei.

Gelöst:

- Verschiedene Fehlerbehebungen.
- Verschiedene Leistungsverbesserungen.

Versionshinweise für AWS SCT Build 640

Die Änderungen für die AWS SCT Versionen 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 und 1.0.640 wurden in der Version 1.0.640 zusammengeführt. AWS SCT

> **Note**
>
> AWS SCT Die Änderungen in Build 1.0.640 gelten für Windows, Ubuntu und Fedora. Sie gelten nicht für macOS.
> Sie können AWS SCT Version 1.0.640 oder höher nicht auf Apple macOS installieren. AWS SCT Version 1.0.632 war die letzte Version, die die Installation auf Apple macOS unterstützte.

In den folgenden Tabellen finden Sie eine Liste der Funktionen und Fehlerbehebungen für die AWS Schema Conversion Tool -Versionen, die in Version 1.0.640 kombiniert wurden. Diese Tabellen gruppieren Funktionen und Fehlerbehebungen anhand der Quell-Engine.

Themen

- Version 1.0.640 Oracle-Änderungen
- Version 1.0.640 Änderungen an Microsoft SQL Server
- Änderungen an MySQL-Version 1.0.640
- Version 1.0.640 PostgreSQL-Änderungen
- Änderungen an der Db2-LUW-Version 1.0.640
- Teradata-Änderungen an Version 1.0.640
- Version 1.0.640 ändert sich für andere Engines

Version 1.0.640 Oracle-Änderungen

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen Oracle die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	Aurora PostgreSQL	Die Konvertierung von SQL-Code in Java- und Pro*C-Anwendungen wurde implementiert.
Oracle	PostgreSQL Aurora PostgreSQL	Die Leistung der folgenden Funktionen bei Verwendung in einer WHERE-Klausel wurde verbessert: • aws_oracle_ext.to_date • aws_oracle_ext.to_char • aws_oracle_ext.to_number • aws_oracle_ext.sysdate • aws_oracle_ext.sys_context
Oracle	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Oracle	PostgreSQL/Aurora PostgreSQL	Unterstützung für DBMS_UTILITY.GET_TIME wurde hinzugefügt. Folgende Emulationen wurden hinzugefügt: • DBMS_UTILITY.GET_TIME • DBMS_UTILITY.FORMAT_CALL_STACK • DBMS_UTILITY.CURRENT_INSTANCE

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	MariaDB/MySQL/Aurora MySQL/Microsoft SQL Server Mode=Oracle/PostgreSQL/Aurora PostgreSQL/RDSOracle	Unterstützung der Freigabeklausel für TABLE(DATA,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) und SEQUENCE(DATA) wurde hinzugefügt.
Oracle	PostgreSQL/Aurora PostgreSQL/OracleRDS	Die DEFAULT-Definition einer Spalte kann so erweitert werden, dass DEFAULT für die explizite NULL-Einfügung angewendet wird. Die DEFAULT-Klausel hat eine neue ON NULL-Klausel. Diese neue Klausel weist die Datenbank an, einen angegebenen Standards paltenwert zuzuweisen, wenn eine INSERT-Anweisung versucht, einen Wert zuzuweisen, der als NULL ausgewertet wird.
Oracle	MariaDB/MariaDB (SQL MODE=ORACLE)	Zusätzliche Unterstützung für „Identitätsspalten“, die zum Zeitpunkt des Einfügens automatisch erhöht werden.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	MariaDB 10.2/MariaDB 10.3/PostgreSQL MySQL/Aurora MySQL/PostgreSQL/Aurora	Die `DEFAULT`-Klausel verfügt über eine neue `ON NULL`-Klausel, die die Datenbank anweist, einen angegebenen Standardspaltenwert zuzuweisen, wenn eine INSERT-Anweisung versucht, einen Wert zuzuweisen, der als `NULL` ausgewertet wird.
Oracle	Oracle RDS/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für `IDENTITY` wurde hinzugefügt.
Oracle	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL/Aurora PostgreSQL	Die Überprüfung von ANYDATA IS NULL/IS NOT NULL mithilfe der Extension Pack-Routine wurde implementiert. Es wurde die Emulation der VALUE-Funktion implementiert, die in einer Abfrage verwendet wird, die auf der TABLE-Funktion von basiert. XMLSequence DBMS_LOB-Unterstützung für die folgenden integrierten Routinen wurde hinzugefügt: • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Amazon Redshift	Konvertierung von Cursorattributen in verschachtelten Blöcken wurde implementiert. Amazon Redshift unterstützt keine Sammlungen. Verwandte Variablen werden als VARCHAR konvertiert. Alle Erfassung svorgänge außer dem Zuweisen einer Variablen zu einer anderen werden abgelehnt, einschließlich des Zugriffs auf Initiations- und Prüfelemente. Amazon Redshift Redshift-Verteilungsstil = AUTO implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL/Aurora PostgreSQL	Wenn ein nicht reserviertes Wort in Oracle in PostgreSQL reserviert ist, gilt Folgendes: • Wenn das Wort in Anführungszeichen gesetzt wird, wird seine Groß- und Kleinschreibung beibehalten. • Wenn das Wort nicht in Anführungszeichen gesetzt wird, wird es in Großbuchstaben umgewandelt und in Anführungszeichen gesetzt. Die Fähigkeit wird implementiert, Funktionen als Eingabe für LTRIM-, RTRIM- und TRIM-Funktionen zu verwenden. SELECT DISTINCT, ORDER BY-Ausdrücke müssen in der Auswahlliste angezeigt werden. Fügt für Cursorparameter, die auf einen Parameter mit einem DEFAULT-Wert folgen, die DEFAULT IS NULL-Klausel AWS SCT hinzu Source OUT-Cursorparameter werden in IN-Cursorparameter konvertiert. Die Paketvariable wird durch Hinzufügen der Option „Paketvariablen Logik-Implementierung“ unter „Konvertierungseinstellungen“ erneut implementiert. Verfügbare Einstellungen sind: „Sitzungsvariablen“ und „plv8 globale Objekte“. Der Standardwert ist „Sitzungsvariablen“. AUTONOUS_TRANSACTION-Pragma-Unterstützung mit dblink und pg_background wurde implementiert.
Oracle	Alle	Ansicht SYS_%_TAB_COMMENTS wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	Variableneingaben für Filter werden in PostgreSQL nicht unterstützt. Wenn bei der Konvertierung von Oracle nach PostgreSQL ein Variablenfilter festgestellt wird, wird nun eine Ausnahme gemeldet.
Oracle	Amazon Redshift	Gespeicherter Code FOR..LOOP und Verbesserungen an Cursorkonvertierung wurden implementiert. Gespeicherter Codeaufruf von Funktionen/Prozeduren mit Standardparametern wurde implementiert. Gespeicherte Code-Fähigkeit zum UPDATE mit Alias ohne WHERE-Klausel wurde implementiert. Gespeicherte Codefunktionen,die zusätzliche Fälle mit SELECT FROM dual bilden, wurden implementiert. Gespeicherter Code Table%ROWTYPE-Parameter und Paketvariablen wurden implementiert. Gespeicherter Code, der von JAVA und externen Prozeduren verwendet wird, wurde implementiert. Standard-Oracle-Paket wurde in gespeichertem Code implementiert.

Version 1.0.640 Änderungen an Microsoft SQL Server

Die folgende Tabelle listet Änderungen an Build 1.0.640 auf, in denen Microsoft SQL Server die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft Azure/Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL/MySQL/	Unterstützung für COLUMN STORE-Indizes wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	Aurora MySQL	
Microsoft SQL Server	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für das Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für Datenbankknoten und Edgetabellentypen wurde hinzugefügt.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für TEMPORAL TABLES wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQ L/MariaDB	Unterstützung für die DML-Verarbeitung für SQL Server Graph Architecture hinzugefügt.
SQL Server	Aurora PostgreSQ L	Die Option, Parameter ohne `par_`-Präfix zu konvertieren, wurde hinzugefügt.
Azure/SQL Server	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefü gt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	AWS Glue (Python-Shell)	Verbesserungen an der Konvertierung, einschließlich: • Integrierte Funktionskonvertierungen wurden in Python.String implementiert. • EXECUTE und EXEC wurden in gespeicherten Code implementiert. • Wurde mithilfe von Tabellentypen implementiert.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Wurde implementiert und macht $TMP-Prozeduren optional.
SQL Server	MySQL/ Aurora MySQL	Erweiterte arithmetische Operationen mit Datumsangaben. Konstruktionsemulation 'TOP (Ausdruck) WITH TIES. Nach dem Aufruf von Prozeduren mit dem generierten Refcursor wird der Refcursor nun geschlossen. Das Festlegen einer GLOBAL-Isolationsebene wird in Aurora MySQL nicht unterstützt. Nur der Sitzungsbereich kann geändert werden. Das Standardverhalten von Transaktionen besteht darin, REPEATABLE READ und konsistente Lesevorgänge zu verwenden. Anwendungen, die für die Verwendung mit READ COMMITED entwickelt wurden, müssen möglicherweise geändert werden. Alternativ können sie den Standardwert explizit in READ COMMITTED ändern.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	AWS Glue (Python-Shell)	SQL Server-Anweisungen generieren eine vollständige Ergebnismenge, aber es gibt Zeiten, in denen die Ergebnisse am besten Zeile für Zeile verarbeitet werden. Das Öffnen eines Cursors in einer Ergebnismenge ermöglicht die zeilenweise Verarbeitung der Ergebnismenge. Sie können einen Cursor einer Variablen oder einem Parameter mit einem Cursordatentyp zuweisen. Implementiert das Einschließen einer Reihe von Transact-SQL-Anweisungen für gespeicherten Code, sodass eine Gruppe von Transact-SQL-Anweisungen ausgeführt werden kann, obwohl Python BEGIN und END von SQL Server nicht unterstützt. control-of-flow Die SQL Server-Anweisungen LABEL und GOTO werden von nicht unterstützt. AWS Glue Wenn AWS SCT ein Label im Code findet, wird es übersprungen. Wenn AWS SCT auf eine GOTO-Anweisung stößt, wird sie kommentiert.
SQL Server	Amazon Redshift	Die bedingte Verarbeitung von Transact-SQL-Anweisungen für den gespeicherten Code durch Implementierung des IF... ELSE-Steuerelements wurde implementiert. Implementiert, das eine Reihe von Transact-SQL-Anweisungen für gespeicherten Code einschließt, sodass eine Gruppe von Transact-SQL-Anweisungen als Block ausgeführt werden kann. Unterstützt verschachtelte BEGIN... END-Blöcke. SET und SELECT wurden in gespeicherten Code implementiert. CREATE INDEX wurde in Amazon Redshift (das keine Indizes unterstützt) implementiert, indem ein benutzerdefinierter Sortierschlüssel für die Tabellen erstellt wurde.

Änderungen an MySQL-Version 1.0.640

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen MySQL die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
MySQL	PostgreSQL 12.x	Unterstützung für generierte Spalten wurde hinzugefügt.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
MySQL	PostgreSQL/Aurora PostgreSQL 11.	Unterstützung für Folgendes wurde hinzugefügt: • Transaktionen wurden in gespeicherte SQL-Prozeduren eingebettet. • Die Möglichkeit, gespeicherte SQL-Prozeduren aufzurufen. • Die Möglichkeit, gespeicherte SQL-Prozeduren zu erstellen.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Version 1.0.640 PostgreSQL-Änderungen

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen PostgreSQL die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
PostgreSQL	MySQL 8.x	MySQL unterstützt jetzt die Erstellung von funktionalen Indexschlüsselteilen, die Ausdruckswerte und nicht Spaltenwerte indizieren. Funktionale Schlüsselteile ermöglichen die Indizierung von Werten, die nicht auf andere Weise indiziert werden können, z. B. JSON-Werte. MySQL unterstützt jetzt CTE und rekursive CTE.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	Unterstützung für Folgendes wurde hinzugefügt: • Transaktionen wurden in gespeicherte SQL-Prozeduren eingebettet. • Die Möglichkeit, gespeicherte SQL-Prozeduren aufzurufen. • Die Möglichkeit, gespeicherte SQL-Prozeduren zu erstellen.
PostgreSQL	MySQL 8.x	MySQL-Unterstützung für absteigende Indizes wurde hinzugefügt. DESC in einer Indexdefinition wird nicht mehr ignoriert, sondern bewirkt, dass Schlüsselwerte in absteigender Reihenfolge gespeichert werden. MySQL-Unterstützung wurde hinzugefügt zur Verwendung von Ausdrücken als Standardwerte in Datentypspezifikationen, einschlie

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		ßlich Ausdrücke als Standardwerte für die Datentypen BLOB, TEXT, GEOMETRY und JSON. Mehrere vorhandene Aggregationsfunktionen können jetzt als Fensterfunktionen verwendet werden. • AVG() • BIT_AND() • BIT_OR() • BIT_XOR() • COUNT() • JSON_ARRAYAGG() • JSON_OBJECTAGG() • MAX() • MIN() • STDDEV_POP() • STDDEV() • STD() • STDDEV_SAMP() • SUM() • VAR_POP() • VARIANCE() • VAR_SAMP() MySQL unterstützt Fensterfunktionen, die für jede Zeile aus einer Abfrage eine Berechnung unter Verwendung zu dieser Zeile gehöriger Zeilen durchführen • CUME_DIST() • DENSE_RANK() • FIRST_VALUE()

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		• LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK() • ROW_NUMBER()
PostgreSQL	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
PostgreSQL/Aurora PostgreSQL	Alle	Systemansicht mit Sysindexes-Emulation wurde hinzugefügt. Wenn eine SELECT-Anweisung in einer Prozedur ohne Angabe von INTO vorhanden ist, wird der Parameter INOUT p_refcur vom Typ refcursor für eine Prozedur am Ziel erstellt.

Änderungen an der Db2-LUW-Version 1.0.640

In der folgenden Tabelle sind die Änderungen in Build 1.0.640 aufgeführt, bei denen DB2 LUW die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
DB2 LUW	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
DB2 LUW	MySQL 8.0.17	Unterstützung für CHECK-Beschränkungen wurde hinzugefügt.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Teradata-Änderungen an Version 1.0.640

In der folgenden Tabelle sind die Änderungen an den Teradata-Quellmodulen in Build 1.0.640 aufgeführt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Unterstützung für die Anweisungen MERGE und QUALIFY wurde hinzugefügt. LOCKING ROWS FOR ACCESS-Klausel wurde von Teradata-Anweisungen entfernt. Unterstützung für CAST-Funktion wurde hinzugefügt.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Teradata	Teradata	Verbesserungen wurden in REGEXP_INSTR() and REGEXP_SUBSTR() implementiert.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Teradata	Alle	Unterstützung wurde für REGEXP_INSTR() und REGEXP_SUBSTR() hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
Teradata	Amazon Redshift	Die Fähigkeit wurde implementiert, den SQL-Code der Quellstruktur je nach Phase in eine einzelne Datei oder mehrere Dateien mit den

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		Einstellungen in den „Projekteinstellungen“, „Als SQL speichern“ und „Übernehmen“, „Dropdown-Liste“, „Einzelne Datei/Mehrere Dateien“ zu speichern. Verbesserungen bei Ansichten und Prozedurkonvertierungen wurden implementiert.
Teradata	Alle	Unterstützung für Teradata-Version 16.20 wurde hinzugefügt

Version 1.0.640 ändert sich für andere Engines

Die folgende Tabelle listet Änderungen an Build 1.0.640 für andere Quell-Engines auf.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Sybase	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
SAP ASE	MariaDB	Folgendes wurde implementiert: • MariaDB 10.4 • EXECUTE IMMEDIATE-Anweisung • DEFAULT-Definitionen • Unterstützung der CHECK-Beschränkungen
SAP ASE	PostgreSQL 12.x	Unterstützung für generierte Spalten wurde hinzugefügt.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	MySQL 8.0.17	Unterstützung für CHECK-Beschränkungen wurde hinzugefügt.
Alle	SQL Server	SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt. SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt. SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.
Vertica	Amazon Redshift	Unterstützung für den Verteilungsstil = AUTO wurde hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
Amazon Redshift	Amazon Redshift	Nicht unterstützte integrierte Funktionen in DML-Anweisungen werden durch NULL als Platzhalter ersetzt.
Sybase	PostgreSQL	Unterstützung für native Funktionen hinzugefügt.
SAP ASE	MySQL/ Aurora MySQL	Die standardmäßige Isolationsebene für Aurora MySQL ist REPEATABLE READ. Das Festlegen einer GLOBAL-Isolationsebene wird in Aurora MySQL nicht unterstützt. Nur der Sitzungsbereich kann geändert werden. Das Standardverhalten von Transaktionen besteht darin, REPEATABLE READ und konsistente Lesevorgänge zu verwenden. Anwendungen, die mit READ COMMITED ausgeführt werden sollen, müssen möglicherweise geändert werden. Sie können auch den Standardwert explizit in READ COMMITTED ändern.
SAP ASE	PostgreSQL	Unterstützung für die CONVERT-Funktion (optimistisch) ohne das Erweiterungspaket hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	Alle	Systemansicht mit Sysindexes-Emulation wurde hinzugefügt. Wenn eine SELECT-Anweisung in einer Prozedur ohne Angabe von INTO vorhanden ist, wird der Parameter INOUT p_refcur vom Typ refcursor für eine Prozedur am Ziel erstellt.
Greenplum	Amazon Redshift	CREATE TEMORATE TABLE wurde wie folgt implementiert:

Neue Funktionen in der AWS Schema Conversion Tool

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Schema Conversion Tool (AWS SCT) -Benutzerhandbuch nach Januar 2018 beschrieben.

Sie können einen RSS-Feed abonnieren, um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten.

Änderung	Beschreibung	Datum
AWS SCT Build #1 .0.672	Build 1.0.672 unterstützt Amazon RDS for PostgreSQL 15 als Ziel und Microsoft SQL Server Version 2022 als Quelle. Es bietet auch Unterstützung für neue Amazon Redshift Redshift-Funktionen im konvertierten Code, implementiert mehrere Konvertierungsverbesserungen für IBM Db2 for z/OS Source und behebt eine Reihe von Konvertierungsproblemen.	8. Mai 2023
AWS SCT Build #1 .0.671	Build 1.0.671 unterstützt Migrationen von Apache Oozie zu. AWS Step Functions Es bietet auch Unterstützung BigQuery als Quelle für den Multiserver-Assessment-Process. Darüber hinaus werden neue Konvertierungseinstellungen für IBM Db2 for z/OS als Quelle hinzugefügt und eine Reihe von Konvertierungsproblemen behoben.	08. März 2023

AWS SCT Build #1 .0.670	Build 1.0.670 unterstützt Migrationen von Hadoop zu Amazon EMR. Es bietet auch Unterstützung für Azure Synapse Analytics als Quelle für den Multiserver-Assessment-Prozess. Darüber hinaus verbessert es die Konvertierung von SQL-Code, der in Java-Anwendungen eingebettet ist, und behebt eine Reihe von Konvertierungsproblemen.	23. Januar 2023
AWS SCT Build #1 .0.669	Build 1.0.669 implementiert die Unterstützung der nativen Partitionierung für die Datenmigration aus Oracle Data Warehouses. Es verbessert auch den Bewertungsprozess für mehrere Server, fügt neue Funktionen für Datenextraktionsagenten hinzu und behebt eine Reihe von Konvertierungsproblemen.	19. Dezember 2022

AWS SCT Build #1 .0.668	Build 1.0.668 implementiert automatische virtuelle Partitionierung für die Datenmigration aus Greenplum-Datenbanken und bietet Unterstützung für die Datenmigration von Snowflake-Datenbanken zu Amazon Redshift. Es verbessert auch die Konvertierung von SQL-Code, der in C#-Anwendungen eingebettet ist, und behebt eine Reihe von Konvertierungsproblemen.	16. November 2022
AWS SCT build #1 .0.667	Build 1.0.667 unterstützt die Informatica Extract, Transform and Load (ETL) Engine als Migrationsquelle. Es aktualisiert auch die Version des Erweiterungspakets, erhöht die unterstützte Mindesttreiberversion für Amazon Redshift und behebt eine Reihe von Konvertierungsproblemen.	13. Oktober 2022

AWS SCT Build #1 .0.666	Build 1.0.666 verbessert die Konvertierung von Java-Anwendungen, indem es Unterstützung für das Framework hinzufügt. MyBatis Es fügt den Erweiterungspaketen auch neue Funktionen hinzu, verbessert den Quellmetadaten-Loader und behebt eine Reihe von Konvertierungsproblemen.	20. September 2022
AWS SCT Build #1 .0.665	Build 1.0.665 bietet Unterstützung für als Migrationsquelle. BigQuery Es implementiert auch die Unterstützung der neuen Version der Babelfish-Funktionskonfigurationsdatei. Darüber hinaus verbessert es die Konvertierung von Data Warehouses zu Amazon Redshift und behebt eine Reihe von Konvertierungsproblemen.	2p. August 2022

AWS SCT Build #1 .0.664	Build 1.0.664 unterstützt Amazon Redshift Serverless als Migrationsquelle oder -ziel. Es implementiert auch den automatischen Speicherausgleich bei Datenextraktionsaufgaben und behebt einen Fehler, bei dem AWS SCT keine Verbindung zu Geräten hergestellt werden konnte. AWS Snowball Edge Darüber hinaus bietet es die Möglichkeit, die Spaltensortierung in Migrationsregeln zu ändern, verbessert die Benutzeroberfläche und behebt eine Reihe von Konvertierungsproblemen.	14. Juli 2022
AWS SCT Build #1 .0.663	Build 1.0.663 bietet Unterstützung für Babelfish for Aurora PostgreSQL 1.2.0 und verbessert die Funktionen für Multiserver-Assessment-Berichte. Es fügt außerdem neue Funktionen in den Migrationsregeln hinzu, behebt zwei Loader-Fehler und behebt eine Reihe von Konvertierungsproblemen.	20. Juni 2022

AWS SCT Build #1 .0.662	Build 1.0.662 implementiert die SQL-Codekonvertierung in C#-Anwendungen und verbessert den Arbeitsab lauf für Bewertungsberichte mit mehreren Servern. Außerdem wurden mehrere Konvertierungsverbesserunge n hinzugefügt und eine Reihe von Konvertierungsproblemen behoben.	19. Mai 2022
AWS SCT Build #1 .0.661	Build 1.0.661 unterstüt zt IBM Db2 for z/OS als Migrationsquelle. Es bietet auch Unterstützung für die Konvertierung von ETL-Skripts (Extract, Transform and Load) in Skripten AWS Glue Studio und behebt eine Reihe von Konvertierungsproblemen.	21. April 2022
AWS SCT Build #1 .0.660	Build 1.0.660 bietet Unterstüt zung für PostgreSQL-Hauptve rsion 14 und MariaDB 10.6 als Migrationsziele. Es bietet auch Unterstützung für die Konvertierung von Oracle-Indizes für materialisierte Ansichten und behebt eine Reihe von Konvertierungsprob lemen.	21. März 2022

AWS SCT Build #1 .0.659	Build 1.0.659 unterstützt die PostgreSQL-Hauptversion 13 auf der Aurora PostgreSQL-Compatible Edition als Migrationsziel. Es implementiert die SQL-Code-Konvertierung in C#-Anwendungen, bietet Unterstützung für Oracle Unified Auditing und behebt eine Reihe von Konvertierungsproblemen.	21. Februar 2022
AWS SCT Build #1 .0.658	Build 1.0.658 ermöglicht die Integration mit der virtuellen Zieldatenbankplattform Amazon Redshift AWS Secrets Manager und bietet zusätzliche Unterstützung. Außerdem wurden eine Reihe von Konvertierungsverbesserungen und Bugfixes hinzugefügt.	20. Januar 2022
AWS SCT Build #1 .0.657	Build 1.0.657 verbessert die Konvertierung von Microsoft SQL Server zu Aurora PostgreSQL-Compatible Edition, Amazon RDS for PostgreSQL und anderen Migrationszielen. Außerdem wurden eine Reihe von Verbesserungen der Benutzeroberfläche und Bugfixes hinzugefügt.	20. Dezember 2021

AWS SCT Build #1 .0.656	Build 1.0.656 bietet Unterstützung für mehrere Quell- und Zieldatenbanken in einem Projekt. Außerdem werden Konvertierung, Optimierungsstrategie und allgemeine Verbesserungen sowie eine Reihe von Fehlerkorrekturen hinzugefügt.	22. November 2021
AWS SCT Build #1 .0.655	Build 1.0.655 implementiert die Konvertierung von FastExport Teradata-Jobskripten nach Amazon Redshift RSQL und erhöht die unterstützte Mindesttreiberversion für Greenplum auf 42.2.5. Außerdem wurden eine Reihe von Verbesserungen und Fehlerkorrekturen hinzugefügt.	18. Oktober 2021
AWS SCT build #1 .0.654	Build 1.0.654 implementiert die Konvertierung von Shell-, Teradata- und Teradata FastLoad Basic Teradata Query (BTEQ) -Skripts in Amazon Redshift RSQL. Es behebt auch eine Reihe von Konvertierungsproblemen und bietet eine Reihe von Verbesserungen und Bugfixes.	16. September 2021

AWS SCT Build #1 .0.653	Build 1.0.653 implementiert die Konvertierung von dynamischem SQL, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde. Es verbessert auch die Konvertierung verschlüsselter Routinen und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.	10. August 2021
AWS SCT Build #1 .0.652	Build 1.0.652 implementiert den Skriptbefehlsmodus in der Befehlszeilenschnittstelle und implementiert Regeln zur Schemaoptimierung. Außerdem wurden eine Reihe von Konvertierungs- und Leistungsverbesserungen sowie Bugfixes hinzugefügt.	30. Juni 2021
AWS SCT Build #1 .0.651	Build 1.0.651 bietet eine Reihe von Verbesserungen und Fehlerkorrekturen. Es bietet auch Zugriff auf die erste Kopie der AWS Schema Conversion Tool CLI-Referenz.	4. Juni 2021

AWS SCT Build #1 .0.650	Build 1.0.650 implementiert die Unterstützung von Amazon RDS for PostgreSQL 13 als Zieldatenbank und aktualisiert Extraktor-Agenten. Es aktualisiert auch die mindestens unterstützte JDBC-Treiberversion für Microsoft SQL Server, Azure und Azure Synapse. Darüber hinaus bietet es eine Reihe von Konvertierungsverbesserungen und Bugfixes.	30. April 2021
AWS SCT Build #1 .0.649	Build 1.0.649 implementiert die Unterstützung von MariaDB 10.5 als Zieldatenbank und implementiert Funktionserweiterungen für die Konvertierung der integrierten Oracle-Funktionen. Es bietet auch eine Reihe von Konvertierungs- und Leistungsverbesserungen sowie Bugfixes.	29. März 2021
AWS SCT Build #1 .0.648	Build 1.0.648 bietet eine Reihe von Konvertierungsverbesserungen und Bugfixes.	22. Februar 2021

AWS SCT Build #1 .0.647	Build 1.0.647 bietet Unterstützung für die Datenbank-Mail-Funktion auf Amazon RDS und implementiert das Laden und Konvertieren von Kommentaren zu Speicherobjekten. Außerdem werden AWS SCT Data Migration Service Assessor und AWS SCT Wizard hinzugefügt und die Baumfilter-Benutzeroberfläche implementiert. Darüber hinaus wurden ein neu gestalteter Abschnitt im Bewertungsbericht sowie eine Reihe von Verbesserungen und Fehlerkorrekturen hinzugefügt.	15. Januar 2021
AWS SCT Build #1 .0.646	Build 1.0.646 bietet Unterstützung für INTERVAL-Datentypen, Identitätsspalten und Cursorkonvertierung sowie eine Reihe von Verbesserungen und Bugfixes.	28. Dezember 2020
AWS SCT Build #1 .0.645	Build 1.0.645 bietet Unterstützung für ETL SSIS bei der AWS Glue Konvertierung sowie eine Reihe von Verbesserungen und Bugfixes.	16. November 2020
AWS SCT Build #1 .0.643-1.0.644	Build 1.0.644 bietet eine Reihe von Konvertierungs-, Leistungs- und Benutzeroberflächenverbesserungen sowie Bugfixes.	14. Oktober 2020

AWS SCT build #1 .0.642	Build 1.0.642 implementiert die Konvertierung von ETL-Paketen von Microsoft SQL Server Integration Services zu AWS Glue und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.	28. August 2020
AWS SCT Build #1 .0.641	SSL-Unterstützung für Datenextraktoren hinzugefügt. Build enthält auch eine Reihe von Verbesserungen und Korrekturen.	17. Juli 2020
AWS SCT baut #1 .0.633-1.0.640	Aktualisierung von JDK 8 auf Amazon Corretto JDK 11. Hinzufügung von Tabellen, die andere Aktualisierungen, Änderungen und Korrekturen identifizieren.	22. Juni 2020
Verfügbarkeit von AWS WQF	AWS SCT stellt das Tool AWS Workload Qualification Framework (AWS WQF) nicht mehr zum Herunterladen bereit.	19. Juni 2020

AWS SCT baut #1 .0.632	SCT UI - Neue Registerkarte hinzugefügt, um Fehler anzuzeigen, die beim Anwenden von Skripten auftreten. Sie können die Quellstruktur nun als SQL speichern, wenn Sie aus SAP ASE konvertieren. Verbesserungen für Konvertierungen in PostgreSQL oder Aurora PostgreSQL oder Redshift.	19. November 2019
AWS SCT Builds #1 .0.631 und #1 .0.630 (kombiniert)	Bessere Unterstützung ROWIDs in Oracle und für Systemobjekte in Microsoft SQL Server und SAP ASE. Bessere Handhabung fehlender Spezifizierer von SQL Server-Schemas. Bessere Unterstützung für Konvertierungen von Greenplum in Redshift. Verbesserte Unterstützung für die Konvertierung von gespeichertem Code bei der Umstellung auf Amazon Redshift, MariaDB, MySQL und PostgreSQL.	30. September 2019

AWS SCT Build #1 .0.629	Unterstützung für gespeicherte Prozeduren für Konvertierungen von Netezza. Verbesserte Unterstützung für Konvertierungen in Amazon Redshift, DynamoDB, MySQL und PostgreSQL. Es wurde Unterstützung für SAP ASE 12.5 als Quelle hinzugefügt.	20. August 2019
AWS SCT Build #1 .0.628	Support für die Serviceemulation für Konvertierungen von DB2 SQL Server und Oracle. Verbesserungen für Konvertierungen zu Amazon Redshift, einschließlich mehr Unterstützung für Cursor und gespeicherte Prozeduren.	22. Juni 2019
AWS SCT Build #1 .0.627	Support für Konvertierungen von SQL Server zu gespeicherten Prozeduren in Amazon Redshift. Verbesserungen für die Konvertierung zu PostgreSQL 11 und MySQL 8.0.	31. Mai 2019
AWS SCT Build #1 .0.626	PostgreSQL 11 und MySQL 8.0 werden jetzt als Ziele unterstützt. SAP ASE 15.5 wird nun als Quelle unterstützt.	26. April 2019

AWS SCT Build #1 .0.625	Zu den Updates gehören die Möglichkeit, Teradata BTEQ zu konvertieren AWS Glue, Unterstützung für Konvertierungen auf MariaDB 10.3 mit Unterstützung für den Oracle-Kompatibilitätsmodus, Unterstützung für SAP ASE 15.7 und Serviceersetzungen zur Emulation fehlender Funktionen.	25. März 2019
AWS SCT Build #1 .0.624	Zu den Updates gehören die Möglichkeit zur Konvertierung von Oracle ETL in und die Unterstützung von Konvertierungen von Microsoft SQL Server, Oracle und IBM Db2 LUW zu Amazon RDS for MariaDB. AWS Glue Wir haben auch Unterstützung für die Konvertierungen von SAP ASE in RDS für MySQL sowie Amazon Aurora mit MySQL-Kompatibilität hinzugefügt. Darüber hinaus haben wir Unterstützung für die Orafce-Erweiterung Orafce bei der Oracle-Konvertierung in PostgreSQL hinzugefügt.	22. Februar 2019

AWS SCT Build #1 .0.623	Die Updates bieten folgende Konvertierungsmöglichkeiten : SAP ASE-Datenbanken sowie die Konvertierung von T-SQL-Skripts, DML und DDL in entsprechenden Code oder Komponenten. Wir haben auch Oracle- und Microsoft SQL Server-Emulatoren für bessere Konvertierungen hinzugefügt.	25. Januar 2019
AWS SCT baue #1 .0.622	Aktualisierungen umfassen das Workload Qualification Framework, das den Workload für eine gesamte Migration, einschließlich Datenbank- und App-Änderungen, analysiert.	20. Dezember 2018
AWS SCT baue #1 .0.621	Aktualisierungen beinhalte n Unterstützung für Aurora PostgreSQL 10 als Ziel und die Möglichkeit zur Migration von Netezza mit externen Tabellenoptionen.	21. November 2018
AWS SCT baue #1 .0.620	Aktualisierungen beinhalten die Fähigkeit, SQL-Skripts zu speichern, sowie Unterstüt zung für globale Oracle-Cursor bei der Migration zu MySQL.	22. Oktober 2018
AWS SCT baue #1 .0.619	Die Aktualisierung unterstützt unter anderem die Migration von Apache Cassandra in DynamoDB und Vertica 9 als Quelle.	20. September 2018

AWS SCT baue #1 .0.618	Zu den Updates gehören erweiterte Bewertungsberichte , Unterstützung für die Konvertierung von Oracle ROWIDs und Unterstützung für benutzerdefinierte SQL Server-Tabellen.	24. August 2018
AWS SCT Build #1 .0.617	Zu den Updates gehören erweiterte Bewertungsberichte , Unterstützung für die Konvertierung von Oracle ROWIDs und Unterstützung für benutzerdefinierte SQL Server-Tabellen.	24. Juli 2018
AWS SCT Build #1 .0.616	Updates beinhalten Unterstützung von RDS bei der Konvertierung von Oracle in Amazon RDS for Oracle und bei der Konvertierung von Oracle-Zeitplanobjekten sowie Unterstützung von Oracle-Jobs, Partitionierung und Db2 LUW Version 10.1.	26. Juni 2018
AWS SCT baue #1 .0.615	Updates beinhalten Unterstützung für SQL Server in PostgreSQL GOTO-Anweisungen, PostgreSQL 10 Partitionierung und Db2 LUW Version 10.1.	24. Mai 2018

AWS SCT baue #1 .0.614	Updates beinhalten Unterstützung von Oracle zu Oracle DB Links, SQL Server in PostgreSQL-Inline-Funktionen und die Emulation von Oracle-Systemobjekten.	25. April 2018
AWS SCT baue #1 .0.613	Updates beinhalten Unterstützung von Db2 LUW, die Konvertierung von SQL* Plus-Dateien und SQL Server Windows-Authentifizierung.	28. März 2018
AWS SCT baue #1 .0.612	Updates beinhalten Unterstützung für benutzerdefinierte Datentypzuweisung, einen Schemavergleich für Oracle 10 und die Umwandlung globaler Variablen von Oracle zu PostgreSQL.	22. Februar 2018
AWS SCT baue #1 .0.611	Updates beinhalten Unterstützung von dynamischen Anweisungen von Oracle zu PostgreSQL, das Öffnen der Protokolldatei durch Auswahl einer Fehlermeldung und die Möglichkeit, Schemas in einer Strukturansicht auszublenden.	23. Januar 2018

Frühere Aktualisierungen

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Schema Conversion Tool (AWS SCT) -Benutzerhandbuch vor Januar 2018 beschrieben.

Version	Änderung	Beschreibung	Datum geändert
1.0.608	FIPS-Endpunktunterstützung für Amazon S3	Sie können jetzt mithilfe von FIPS-Endpunkten eine Verbindung AWS SCT zu Amazon S3 und Amazon Redshift beantragen, um die Sicherheitsanforderungen des Federal Information Processing Standard zu erfüllen. Weitere Informationen finden Sie unter AWS Anmeldeinformationen speichern.	17. November 2017
1.0.607	FIPS-Endpunktunterstützung für Amazon S3	Sie können jetzt mithilfe von FIPS-Endpunkten eine Verbindung AWS SCT zu Amazon S3 und Amazon Redshift beantragen, um die Sicherheitsanforderungen des Federal Information Processing Standard zu erfüllen. Weitere Informationen finden Sie unter AWS Anmeldeinformationen speichern.	30. Oktober 2017
1.0.607	Datenextraktionsaufgaben können ignoriert werden LOBs	Wenn Sie Datenextraktionsaufgaben erstellen, können Sie jetzt festlegen, dass große Objekte (LOBs) ignoriert werden, um die Menge der zu extrahierenden Daten zu reduzieren. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen.	30. Oktober 2017
1.0.605	Zugriff auf das Aufgabenprotokoll des Agenten für die Datenextraktion	Sie können jetzt über einen praktischen Link auf der AWS Schema Conversion Tool Benutzeroberfläche auf das Taskprotokoll des Datenextraktionsagenten zugreifen. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen.	28. August 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.604	Converter-Erweiterungen	Die AWS Schema Conversion Tool Engine wurde verbessert und bietet nun verbesserte Konvertierungen für heterogene Migrationen.	24. Juni 2017
1.0.603	Filter für den Agenten der Datenextraktion	Sie können jetzt die Daten filtern, die die Agenten für die Datenextraktion aus dem Data Warehouse extrahieren. Weitere Informationen finden Sie unter Regeln für die Datenmigration erstellen in AWS SCT.	16. Juni 2017
1.0.603	AWS SCT unterstützt zusätzliche Data Warehouse-Versionen	Sie können jetzt das verwenden AWS Schema Conversion Tool , um Ihre Teradata 13- und Oracle Data Warehouse 10-Schemas in entsprechende Amazon Redshift Redshift-Schemas zu konvertieren. Weitere Informationen finden Sie unter Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT.	16. Juni 2017
1.0.602	Unterstützung zusätzlicher Data Warehouses durch die Agenten für die Datenextraktion	Sie können jetzt mit Agenten der Datenextraktion Daten aus Ihren Microsoft SQL Server Data Warehouses extrahieren. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.	11. Mai 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.602	Datenextraktionsagenten können Daten nach Amazon Redshift kopieren	Die Agenten für die Datenextraktion unterstützen jetzt drei Modi für das Hochladen. Sie können jetzt angeben, ob Sie nur Ihre Daten extrahieren, Ihre Daten extrahieren und einfach auf Amazon S3 hochladen oder Ihre Daten direkt in Amazon Redshift extrahieren, hochladen und kopieren möchten. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen.	11. Mai 2017
1.0.601	AWS SCT unterstützt zusätzliche Data Warehouses	Sie können jetzt die verwenden AWS Schema Conversion Tool , um Ihre Vertica- und Microsoft SQL Server-Schemas in entsprechende Amazon Redshift Redshift-Schemas zu konvertieren. Weitere Informationen finden Sie unter Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT.	18. April 2017
1.0.601	Unterstützung zusätzlicher Data Warehouses durch die Agenten für die Datenextraktion	Sie können jetzt mit Agenten der Datenextraktion Daten aus Ihren Greenplum-, Netezza- und Vertica-Data Warehouses extrahieren. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.	18. April 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.601	Unterstützung zusätzlicher Betriebssysteme durch die Agenten für die Datenextraktion	Sie können nun Agenten für die Datenextraktion auf Computern installieren, auf denen die Betriebssysteme Microsoft Windows oder macOS ausgeführt werden. Weitere Informationen finden Sie unter Installation von Extraktionsagenten.	18. April 2017
1.0.601	Datenextraktionsagenten laden automatisch auf Amazon S3 hoch	Datenextraktionsagenten laden Ihre extrahierten Daten jetzt automatisch auf Amazon S3 hoch. Weitere Informationen finden Sie unter Ausgabe der Aufgabe zur Datenextraktion.	18. April 2017
1.0.600	Agenten für die Datenextraktion	Sie können jetzt Datenextraktionsagenten installieren, die Daten aus Ihrem Data Warehouse extrahieren und für die Verwendung mit Amazon Redshift vorbereiten. Sie können den verwenden AWS Schema Conversion Tool , um die Agenten zu registrieren und Datenextraktionsaufgaben für sie zu erstellen. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift mit AWS Schema Conversion Tool.	16. Februar 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.600	Kunden-Feedback	Sie können jetzt Feedback zum AWS Schema Conversion Tool bereitstellen. Sie können einen Fehlerbericht übermitteln, eine Funktionalitätsanforderung stellen oder allgemeine Informationen bereitstellen. Weitere Informationen finden Sie unter Feedback geben.	16. Februar 2017
1.0.502	Integration mit AWS DMS	Sie können den jetzt verwenden AWS Schema Conversion Tool , um AWS DMS Endpunkte und Aufgaben zu erstellen. Sie können die Aufgaben von AWS SCT ausführen und überwachen. Weitere Informationen finden Sie unter Integrieren AWS Database Migration Service mit AWS Schema Conversion Tool.	20. Dezember 2016
1.0.502	Amazon Aurora mit PostgreSQL-Kompatibilität als Zieldatenbank	Das AWS Schema Conversion Tool unterstützt jetzt Amazon Aurora mit PostgreSQL-Kompatibilität als Zieldatenbank. Weitere Informationen finden Sie unter Datenbankschemas konvertieren in AWS Schema Conversion Tool.	20. Dezember 2016
1.0.502	Profilunterstützung	Sie können jetzt verschiedene Profile im AWS Schema Conversion Tool speichern und unkompliziert zwischen diesen wechseln. Weitere Informationen finden Sie unter Profile verwalten in AWS Schema Conversion Tool.	20. Dezember 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.501	Support für Greenplum Database und Netezza	Sie können das jetzt verwenden AWS Schema Conversion Tool , um Ihre Data Warehouse-Schemas von Greenplum Database und Netezza nach Amazon Redshift zu konvertieren. Weitere Informationen finden Sie unter Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT.	17. November 2016
1.0.501	Redshift-Optimierung	Sie können das jetzt verwenden AWS Schema Conversion Tool , um Ihre Amazon Redshift Redshift-Datenbanken zu optimieren. Weitere Informationen finden Sie unter Daten aus Amazon Redshift konvertieren mit AWS Schema Conversion Tool.	17. November 2016
1.0.500	Zuweisen von Regeln	Bevor Sie Ihr Schema mit dem konvertieren AWS Schema Conversion Tool, können Sie jetzt Regeln einrichten, die den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern. Weitere Informationen finden Sie unter Migrationsregeln erstellen in AWS SCT.	4. Oktober 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.500	Wechsel zur Cloud	Sie können jetzt das verwenden AWS Schema Conversion Tool , um Ihr vorhandenes lokales Datenbankschema in eine Amazon RDS-DB-Instance zu kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind. Weitere Informationen finden Sie unter Verwendung des Bewertungsberichts in der AWS Schema Conversion Tool.	4. Oktober 2016
1.0.400	Konvertieren von Data Warehouse-Schemata	Sie können das jetzt verwenden AWS Schema Conversion Tool , um Ihre Data Warehouse-Schemas von Oracle und Teradata nach Amazon Redshift zu konvertieren. Weitere Informationen finden Sie unter Konvertieren von Data Warehouse-Schemas in Amazon RDS mithilfe AWS SCT.	13. Juli 2016
1.0.400	Konvertierung des Anwendungs-SQL-Codes	Sie können jetzt mit dem AWS Schema Conversion Tool SQL-Code in Ihren C++-, C #-, Java- oder anderen Anwendungscode konvertieren. Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren mit AWS SCT.	13. Juli 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.400	Neues Feature	Das enthält AWS Schema Conversion Tool jetzt ein Erweiterungspaket und einen Assistenten, mit denen Sie AWS Lambda Funktionen und Python-Bibliotheken für E-Mail, Jobplanung und andere Funktione n installieren, erstellen und konfigurieren können. Weitere Informationen erhalten Sie unter Verwenden der AWS Lambda Funktionen aus dem AWS SCT Erweiteru ngspaket und Verwenden von benutzerd efinierten Bibliotheken für AWS SCT Erwei terungspakete.	13. Juli 2016
1.0.301	SSL Support	Sie können jetzt mithilfe von Secure Sockets Layer (SSL) eine Verbindung mit Ihrer Quell-Datenbank herstellen, wenn Sie das AWS Schema Conversion Tool verwenden.	19. Mai 2016
1.0.203	Neues Feature	Es wurde eine Unterstützung für MySQL und PostgreSQL als Quelldatenbanken für Konvertierungen hinzugefügt.	11. April 2016
1.0.202	Wartungsv ersion	Unterstützt die Bearbeitung des konvertie rten SQL-Codes, der für die Zieldatenbank-Engine generiert wurde. Implementiert verbesserte Auswahlmöglichkeiten in den Strukturansichten der Quelldatenbank und der Ziel-DB-Instance. Fügt eine Unterstüt zung für die Verbindung mit einer Oracle-Quelldatenbank über TNS (Transparent Network Substrate)-Namen hinzu.	2. März 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.200	Wartungsversion	Implementiert die Unterstützung von PostgreSQL als Zieldatenbank-Engine hinzu. Ermöglicht das Generieren konvertierter Schemata als Skripts und das Speichern dieser Skripts in Dateien, ehe das Schema auf die Ziel-DB-Instance angewendet wird.	14. Januar 2016
1.0.103	Wartungsversion	Implementiert eine Offline-Projektfunktionalität, die Möglichkeit, nach neuen Versionen zu suchen, und eine Speicher- und Leistungsverwaltung.	2. Dezember 2015
1.0.101	Wartungsversion	Fügt den Assistenten Create New Database Migration Project hinzu. Fügt die Möglichkeit zum Speichern des Bewertungsberichts zur Datenbankmigration als PDF-Datei hinzu.	19. Oktober 2015
1.0.100	Vorversion	Stellt das Benutzerhandbuch für die AWS Schema Conversion Tool Vorschauversion bereit.	7. Oktober 2015

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.