aws

Unable to locate subtitle

AWS Glue DataBrew開發人員指南

AWS Glue DataBrew開發人員指南: ***Unable to locate subtitle***

Table of Contents

什麼是AWS Glue DataBrew？

AWS Glue DataBrew是一種視覺化資料準備工具，可讓使用者清除和標準化資料，而無需撰寫任何程式碼。相較於自訂開發的資料準備，使用 DataBrew 有助於將準備資料用於分析和機器學習 (ML) 所需的時間縮短高達 80%。您可以從超過 250 個現成轉換中選擇自動化資料準備任務，例如篩選異常、將資料轉換為標準格式，以及更正無效的值。

使用 DataBrew，商業分析師、資料科學家和資料工程師可以更輕鬆地協作，從原始資料中取得洞見。由於 DataBrew 是無伺服器，無論您的技術層級為何，您都可以探索和轉換數 TB 的原始資料，而不需要建立叢集或管理任何基礎設施。

透過直覺式 DataBrew 界面，您可以互動方式探索、視覺化、清理和轉換原始資料。DataBrew 提供智慧建議，協助您識別難以找到且耗時修正的資料品質問題。透過 DataBrew 準備資料，您可以利用時間對結果採取行動，並更快速地迭代。您可以將轉換儲存為配方中的步驟，稍後可以更新或重複使用其他資料集，並持續部署。

下圖顯示 DataBrew 如何在高階運作。

若要使用 DataBrew，您可以建立專案並連線至您的資料。在專案工作區中，您會在類似網格的視覺化界面中看到資料。在這裡，您可以探索資料並查看價值分佈和圖表，以了解其設定檔。

若要準備資料，您可以從超過 250 point-and-click轉換中進行選擇。這些包括移除 null、取代遺失值、修正結構描述不一致、根據函數建立資料欄等等。您也可以使用轉換來套用自然語言處理 (NLP) 技

術，將句子分割為片語。立即預覽會顯示轉換前後的一部分資料，因此您可以在將其套用至整個資料集之前修改配方。

DataBrew 在資料集上執行配方後，輸出會儲存在 Amazon Simple Storage Service (Amazon S3) 中。清除且準備好的資料集位於 Amazon S3 之後，另一個資料儲存或資料管理系統就可以擷取該資料集。

的核心概念和術語AWS Glue DataBrew

您可以在下面找到核心概念和術語的概觀AWS Glue DataBrew。閱讀本節後，請參閱 入門AWS Glue DataBrew，其中會逐步引導您建立專案、連接資料集和執行任務的程序。

主題

- 專案
- 資料集
- Recipe
- 任務
- 資料沿襲
- 資料設定檔

專案

DataBrew 中的互動式資料準備工作區稱為專案。使用資料專案，您可以管理相關項目的集合：資料、轉換和排程程序。在建立專案的過程中，您可以選擇或建立要處理的資料集。接著，您可以建立配方，這是您希望 DataBrew 採取行動的一組指示或步驟。這些動作會將原始資料轉換為可供資料管道使用的形式。

資料集

資料集僅代表一組資料，即分為資料欄或欄位的資料列或記錄。當您建立 DataBrew 專案時，您可以連線到或上傳要轉換或準備的資料。DataBrew 可以處理來自任何來源、從格式化檔案匯入的資料，並直接連接到不斷增長的資料存放區清單。

對於 DataBrew，資料集是資料的唯讀連線。DataBrew 會收集一組描述性中繼資料來參考資料。DataBrew 無法更改或存放實際資料。為了簡化，我們使用資料集來參考 DataBrew 使用的實際資料集和中繼資料。

Recipe

在 DataBrew 中，配方是一組指示或步驟，用於您希望 DataBrew 採取行動的資料。配方可以包含許多步驟，而每個步驟可以包含許多動作。您可以使用工具列上的轉換工具來設定要對資料進行的所有變更。稍後，當您準備好查看配方的成品時，請將此任務指派給 DataBrew 並進行排程。DataBrew 會儲存有關資料轉換的指示，但不會儲存您的任何實際資料。您可以在其他專案中下載和重複使用配方。您也可以發佈多個版本的配方。

任務

DataBrew 會執行您在建立配方時設定的指示，以執行轉換資料的任務。執行這些指示的程序稱為任務。任務可以根據預設排程，將您的資料配方付諸行動。但您不會受限於排程。您也可以隨需執行任務。如果您想要分析一些資料，則不需要配方。在這種情況下，您可以直接設定設定檔任務來建立資料設定檔。

資料沿襲

DataBrew 會在視覺化界面中追蹤您的資料，以判斷其原始伺服器，稱為資料歷程。此檢視顯示資料如何流經來自原始來源的不同實體。您可以看到其原始伺服器、受到其影響的其他實體、隨著時間的推移發生了什麼情況，以及存放的位置。

資料設定檔

當您描述資料時，DataBrew 會建立名為資料描述檔的報告。此摘要會告訴您資料的現有形狀，包括內容的內容、資料結構及其關係。您可以執行資料設定檔任務，為任何資料集建立資料設定檔。

產品和服務整合

使用本節了解哪些 產品和服務與 DataBrew 整合。

DataBrew 可搭配下列AWS服務進行聯網、管理和控管：

- Amazon CloudFront
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS Step Functions

DataBrew 適用於下列AWS資料湖和資料存放區：

- AWS Lake Formation
- Amazon S3

DataBrew 支援下列檔案格式和擴充功能來上傳資料。

Format (格式)	副檔名 （選用）	壓縮檔案的延伸模組 （必要）

逗號分隔值	`.csv`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Microsoft Excel 工作手冊	`.xlsx`	無壓縮支援

JSON (JSON 文件和 JSON 行）	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Apache ORC	`.orc`	`.zlib` `.snappy`

Apache Parquet	`.parquet`	`.gz`
		`.snappy`
		`.lz4`

DataBrew 會將輸出檔案寫入 Amazon S3，並支援下列檔案格式和副檔名。

Format (格式)	副檔名（未壓縮）	副檔名（壓縮）
逗號分隔值	.csv	.csv.snappy，.csv.gz, .csv.lz4, csv.bz2, .csv.deflate，csv.br
標籤分隔值	.csv	.tsv.snappy，.tsv.gz, .tsv.lz4, tsv.bz2, .tsv.deflate，tsv.br
Apache Parquet	.parquet	.parquet.snappy，.parquet.gz，.parquet.lz4，.parquet.lzo，.parquet.br
AWS Glue Parquet	不支援	.glue.parquet.snappy
Apache Avro	.avro	.avro.snappy，.avro.gz, .avro.lz4，.avro.bz2，.avro.deflate，.avro.br
Apache ORC	.orc	.orc.snappy，.orc.lzo, .orc.zlib
XML	.xml	.xml.snappy，.xml.gz, .xml.lz4, .xml.bz2, .xml.deflate，.xml.br
JSON（僅限 JSON 行格式）	.json	.json.snappy，.json.gz, .json.lz4，json.bz2, .json.deflate，.json.br
Tableau Hyper	不支援	不適用

設定AWS Glue DataBrew

開始使用 之前AWS Glue DataBrew，您需要設定一些許可、使用者和角色。首先執行下列步驟：

1. 視需要註冊AWS帳戶，並建立AWS Identity and Access Management(IAM) 政策以讓使用者執行 DataBrew：
   - 註冊新AWS帳戶並新增使用者。如需詳細資訊，請參閱設定新AWS帳戶。
   - 為主控台使用者新增 IAM 政策。 具有這些許可的使用者可以在 上存取 DataBrew AWS 管理主控台。
   - 新增 IAM 角色資料資源的許可。 具有這些許可的 IAM 角色可以代表使用者存取資料。

   您必須是 IAM 管理員才能建立使用者、角色和政策。
2. 新增 DataBrew 的使用者或群組。已連接正確許可的使用者或群組可以在 主控台上存取 DataBrew。
3. 新增具有存取 DataBrew 資料許可的角色。具有正確許可的角色可以代表使用者存取資料。

設定新AWS帳戶

如果您沒有AWS帳戶，請註冊AWS帳戶並建立 IAM 管理員使用者。

如果您沒有AWS 帳戶，請完成下列步驟來建立一個。

註冊AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

若要建立管理員使用者，請選擇下列其中一個選項。

選擇一種管理管理員的方式	到	根據	您也可以
在 IAM Identity Center (建議)	使用短期憑證存取AWS。 這與安全性最佳實務一致。有關最佳實務的資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。	請遵循 AWS IAM Identity Center使用者指南的入門中的說明。	在 AWS Command Line Interface使用者指南中設定AWS CLI以使用 來設定AWS IAM Identity Center程式設計存取。
在 IAM 中 (不建議使用)	使用長期憑證存取AWS。	請遵循《IAM 使用者指南》中建立 IAM 使用者以進行緊急存取 的指示。	請依照《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰設定以程式設計方式存取。

如需詳細資訊，請參閱《IAM 使用者指南》中的以下主題：

- 什麼是 IAM？
- 使用 IAM 進行設定
- 建立管理使用者和群組（主控台）

設定AWS CLI

如果您計劃使用 JupyterLab 或 DataBrew API，請務必安裝AWS Command Line Interface(AWS CLI)。您不需要使用 DataBrew 主控台或執行入門練習中的步驟。

若要設定AWS CLI

1. AWS CLI使用下列步驟下載並設定：

   - 安裝AWS CLI
   - 組態基本概念

2. 在命令提示中輸入下列 DataBrew 命令來驗證設定。

```
aws databrew help
```

如果此陳述式傳回錯誤 "aws: error: argument command: Invalid choice"，後面接著長的服務清單，請解除安裝AWS CLI，然後重新安裝。此動作不會覆寫您現有的組態。

AWS CLI命令會使用組態中的預設AWS區域，除非您使用參數或設定檔來設定它。您可以將 `--region` 參數新增至每個命令。

如果您願意，可以在 `~/.aws/config`或 `%UserProfile%/.aws/config`（在 Microsoft Windows 上）中新增具名設定檔。具名設定檔也可以保留其他設定，如下列範例所示。

```
[profile databrew]
aws_access_key_id = ACCESS-KEY-ID-OF-IAM-USER
aws_secret_access_key = SECRET-ACCESS-KEY-ID-OF-IAM-USER
region = us-east-1
output = text
```

設定AWS Identity and Access Management(IAM) 許可

開始使用之前，您需要在 IAM 中設定一些項目。您必須是 管理員或從中取得協助。不過，如果您的 帳戶具有管理員存取權，則可以自行執行這些任務。您可以在本節中找到每個任務的簡單說明。

以下是您需要執行的操作概觀：

- 在此程序中，您會新增使用者。您不需要新增使用者，您可以使用現有的使用者。您可以連接 DataBrew 許可，讓使用者可以開啟 DataBrew 主控台。
- 建立 IAM 角色。角色允許特定動作，並在使用時在限制內提供許可。例如，它僅適用於您AWS帳戶中的使用者。您可以稍後新增更多限制。
- 建立您需要的 IAM 政策。政策是允許使用者執行的物件清單。若要建立政策，您可以開啟另一個主控台頁面，並從您下載的檔案貼上文字。

> **Note**
>
> 我們在這裡提供的基本設定資訊。我們建議您花一些時間自訂許可，以滿足安全和合規需求。如果您需要協助，請聯絡您的管理員或AWS Support。

新增必要的許可

1. 建立 IAM 政策以讓使用者執行 DataBrew，方法如下：

   - 為主控台使用者新增自訂 IAM 政策。如果您不需要自訂政策，您可以改為選擇AWS受管政策。只需在步驟 2 中將其新增至使用者即可。具有這些許可的使用者可以存取 DataBrew 服務主控台。
   - 新增資料資源的許可。具有這些許可的 IAM 角色可以代表使用者存取資料。

   您必須是管理員才能建立使用者、角色和政策。
2. 新增 DataBrew 的使用者或群組。已連接正確許可的使用者或群組可以存取 DataBrew 主控台。
3. 新增具有存取 DataBrew 資料許可的角色。具有正確許可的角色可以代表使用者存取資料。

設定 DataBrew 的 IAM 政策

您可以使用 IAM 政策來管理許可。政策可讓您更輕鬆地一次新增相關的許可，而不是一次新增一個許可。

我們建議您使用我們提供的相同名稱來建立政策。我們會在整份文件中針對這些政策使用下列顯示的名稱。如果您需要聯絡AWS Support，使用這些名稱也會讓您更輕鬆。不過，您可以選擇同時變更政策名稱及其內容。如需 IAM 政策的詳細資訊，請參閱《IAM 使用者指南》中的建立客戶受管政策。

建立使用 DataBrew 所需的政策後，您可以將它們連接到使用者和角色。本節稍後將介紹如何執行此操作。

主題

為主控台使用者新增 IAM 政策

為 使用者設定許可AWS 管理主控台是選用的，但如果您需要主控台存取，請先執行此步驟。

若要在主控台上設定存取 DataBrew 的許可，請選擇下列其中一項：

- 使用由 管理的政策AWS：AwsGlueDataBrewFullAccessPolicy。如果您選擇此選項，請跳到下一個政策 新增 IAM 角色資料資源的許可。
- 建立本節所述的政策 AwsGlueDataBrewCustomUserPolicy。此選項可讓您使用其他自訂安全需求來自訂政策。

下列政策會授予執行 DataBrew 主控台所需的許可。您可以使用 IAM 提供這些許可。

定義 DataBrew 的 AwsGlueDataBrewCustomUserPolicy IAM 政策（主控台）

1. 下載 IAM AwsGlueDataBrewCustomUserPolicy 政策的 JSON。
2. 登入AWS 管理主控台並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
3. 在導覽窗格中，選擇政策。
4. 針對每個政策，選擇建立政策。
5. 在建立政策畫面上，導覽至 JSON 索引標籤。
6. 複製您下載的政策 JSON 陳述式。將它貼到編輯器中的範例陳述式上。
7. 確認政策已根據您的 帳戶、安全需求和所需AWS資源進行自訂。如果您需要進行變更，您可以在編輯器中進行變更。
8. 選擇檢閱政策。

定義 DataBrew 的 AwsGlueDataBrewCustomUserPolicy IAM 政策 (AWS CLI)

1. 下載 IAM AwsGlueDataBrewCustomUserPolicy 政策的 JSON。
2. 依照先前程序的第一個步驟所述自訂政策。
3. 執行下列命令來建立政策。

```
aws iam create-policy --policy-name AwsGlueDataBrewCustomUserPolicy --policy-
document file://iam-policy-AwsGlueDataBrewCustomUserPolicy.json
```

新增 IAM 角色資料資源的許可

若要連線至資料，AWS Glue DataBrew需要具有可代表使用者傳遞的 IAM 角色。您可以在下面找到如何建立稍後連接到 IAM 角色的政策。

此AwsGlueDataBrewDataResourcePolicy政策會授予使用 DataBrew 連線至資料所需的許可。對於存取其他AWS資源中資料的任何操作，例如存取 Amazon S3 中的物件，DataBrew 需要代表您存取資源的許可。

定義 DataBrew 的 AwsGlueDataBrewDataResourcePolicy IAM 政策（主控台）

1. 下載適用於 的 JSONAwsGlueDataBrewDataResourcePolicy。
2. 登入AWS 管理主控台，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
3. 在導覽窗格中，選擇政策。
4. 針對每個政策，選擇建立政策。
5. 在建立政策畫面上，導覽至 JSON 索引標籤。
6. 複製您下載的政策 JSON 陳述式。將它貼到編輯器中的範例陳述式上。
7. 確認政策已根據您的 帳戶、安全需求和所需AWS資源進行自訂。如果您需要進行變更，您可以在編輯器中進行變更。
8. 選擇檢閱政策。

定義 DataBrew 的 AwsGlueDataBrewDataResourcePolicy IAM 政策 (AWS CLI)

1. 下載適用於 的 JSONAwsGlueDataBrewDataResourcePolicy。
2. 依照先前程序的第一個步驟所述自訂政策。
3. 執行下列命令來建立政策。

```
aws iam create-policy --policy-name AwsGlueDataBrewDataResourcePolicy --policy-
document file://iam-policy-AwsGlueDataBrewDataResourcePolicy.json
```

設定 DataBrew 的 IAM 政策

您可以在下面找到與 DataBrew 搭配使用之 IAM 政策的詳細資訊和範例。此處提供基本政策的詳細資訊。此外，還有更多不需要使用 DataBrew 的範例。它們是您可以在特定情況下使用的其他組態。

主題

- AwsGlueDataBrewCustomUserPolicy
- AwsGlueDataBrewDataResourcePolicy
- 搭配 DataBrew 使用 Amazon S3 物件的 IAM 政策

- 搭配 DataBrew 使用加密的 IAM 政策

AwsGlueDataBrewCustomUserPolicy

此AwsGlueDataBrewCustomUserPolicy政策會授予使用 DataBrew 主控台所需的大部分許可。此政策中指定的部分資源是指 DataBrew 所使用的服務。這些包括 的名稱AWS Glue Data Catalog、Amazon S3 儲存貯體、Amazon CloudWatch Logs 和資源AWS KMS。它類似於名為 的AWS受管政策AwsGlueDataBrewFullAccessPolicy。

下表說明此政策授予的許可。

Action	Resource	Description
"databrew:*"	"*"	准許執行所有 DataBrew API 操作。
"glue:GetDatabases" "glue:GetPartitions" "glue:GetTable" "glue:GetTables" "glue:GetDataCatalogEncryptionSettings"	"*"	允許列出AWS Glue資料庫和資料表。
"dataexchange:ListDataSets" "dataexchange:ListDataSetRevisions" "dataexchange:ListRevisionAssets" "dataexchange:CreateJob" "dataexchange:StartJob"	"*"	允許列出資料集中的AWS Data Exchange 資源。

Action	Resource	Description
"dataexchange:GetJob"		
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	允許列出用於任務輸出加密的AWS KMS金鑰。
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	允許加密任務輸出。
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionConfiguration"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許列出專案、資料集和任務的 Amazon S3 儲存貯體。允許將輸出檔案傳送至 S3。
"sts:GetCallerIdentity"	"*"	取得目前發起人的相關資訊。
"cloudtrail:LookupEvents",	"*"	允許列出資料集AWS CloudTrail的事件（資料歷程）。
"iam:ListRoles" "iam:GetRole"	"*"	允許列出用於專案和任務的 IAM 角色。

AwsGlueDataBrewDataResourcePolicy

此AwsGlueDataBrewDataResourcePolicy政策會授予連線至資料和設定 DataBrew 所需的許可。

下表說明此政策授予的許可。

Action	Resource	Description
"s3:GetObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	可讓您預覽檔案。
"s3:PutObject" "s3:PutBucketCORS"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許將輸出檔案傳送至 S3。
"s3:DeleteObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許刪除 DataBrew 建立的物件。
"s3:ListBucket"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許從專案、資料集和任務列出 Amazon S3 儲存貯體。
"kms:Decrypt"	"arn:aws:kms:::key/key_ids"	允許解密加密的資料集。
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	允許加密任務輸出。
"ec2:DescribeVpcEndpoints" "ec2:DescribeRouteTables" "ec2:DeleteNetworkInterface" "ec2:DescribeNetworkInterfaces"	"*"	允許在執行任務和專案時設定 Amazon EC2 網路項目，例如虛擬私有雲端 (VPCs)。

Action	Resource	Description
"ec2:DescribeSecurityGroups" "ec2:DescribeSubnets" "ec2:DescribeVpcAttribute" "ec2:CreateNetworkInterface"		
"ec2:DeleteNetworkInterface"	"*"	允許刪除 VPC 中的網路界面。
"ec2:CreateTags" "ec2:DeleteTags"	"arn:aws:ec2:::network-interface/*", "arn:aws:ec2:::security-group/*"	允許建立和刪除標籤。 如果您在啟用 VPC 的情況下使用AWS Glue Data Catalog，則需要這些許可。DataBrew 會將資料傳遞至AWS Glue以執行您的任務和專案。這些許可允許標記為開發端點建立的 Amazon EC2 資源。使用AWS Glue標記 Amazon EC2 網路介面、安全群組和執行個體aws-glue-service-resource 。
"logs:CreateLogGroup" "logs:CreateLogStream" "logs:PutLogEvents"	"arn:aws:logs:::log-group:/aws-glue-databrew/*"	允許將日誌寫入 Amazon CloudWatch Logs DataBrew 會將日誌寫入名稱開頭為 的日誌群組aws-glue-databrew 。

Action	Resource	Description
`"lakeformation:GetDataAccess"`	`"*"`	允許存取AWS Lake Formation，也允許`"Glue":"GetTable"` 提供 使用 Lake Formation 需要在 Lake Formation 主控台中進一步設定。

搭配 DataBrew 使用 Amazon S3 物件的 IAM 政策

此`AwsGlueDataBrewSpecificS3BucketPolicy`政策會授予代表非管理使用者存取 S3 所需的許可。

自訂政策，如下所示：

1. 取代政策中的 Amazon S3 路徑，使其指向您要使用的路徑。在範例文字中，*`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`*代表特定物件或檔案。*`BUCKET-NAME-2/`*代表路徑名稱開頭為 的所有物件 (*)`BUCKET-NAME-2/`。更新這些項目以命名您正在使用的儲存貯體。
2. （選用）在 Amazon S3 路徑中使用萬用字元，以進一步限制許可。如需詳細資訊，請參閱「IAM 使用者指南」中的 IAM 政策元素：變數和標籤。

   安全最佳實務：若要防止未經授權存取其他AWS帳戶中具有類似名稱的 Amazon S3 儲存貯體，請在政策中包含 `aws:ResourceAccount`條件金鑰。這可確保 DataBrew 只能存取您AWS帳戶中的儲存貯體，即使使用萬用字元資源 ARNs 也是如此。將下列條件新增至您的政策陳述式：

   ```
   "Condition": {
   "StringEquals": {
   "aws:ResourceAccount": "123456789012"
   }
   }
   ```

   `123456789012` 以您的實際AWS帳戶 ID 取代 。

   為此，您可以限制動作 `s3:PutObject`和 的許可`s3:PutBucketCORS`。只有建立 DataBrew 專案的使用者才需要這些動作，因為這些使用者需要能夠將輸出檔案傳送至 S3。

如需詳細資訊並查看您可以新增至 Amazon S3 IAM 政策的一些範例，請參閱《Amazon S3 開發人員指南》中的儲存貯體政策範例。 Amazon S3

下表說明此政策授予的許可。

Action	Resource	Description
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	可讓您預覽檔案。
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	允許將輸出檔案傳送至 S3。
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	允許刪除物件。

定義 DataBrew 的 AwsGlueDataBrewSpecificS3BucketPolicy IAM 政策（主控台）

1. 下載 IAM `AwsGlueDataBrewSpecificS3BucketPolicy` 政策的 JSON。
2. 登入AWS 管理主控台並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
3. 在導覽窗格中，選擇政策。
4. 針對每個政策，選擇建立政策。
5. 在建立政策畫面上，導覽至 JSON 索引標籤。
6. 將政策 JSON 陳述式貼到編輯器中的範例陳述式上。
7. 確認政策已根據您的 帳戶、安全需求和所需AWS資源進行自訂。如果您需要進行變更，您可以在編輯器中進行變更。
8. 選擇檢閱政策。

定義 DataBrew 的 AwsGlueDataBrewSpecificS3BucketPolicy IAM 政策 (AWS CLI)

1. 下載適用於 的 JSONAwsGlueDataBrewSpecificS3BucketPolicy。
2. 依照先前程序的第一個步驟所述自訂政策。
3. 執行下列命令來建立政策。

```
aws iam create-policy --policy-name AwsGlueDataBrewSpecificS3BucketPolicy --policy-
document file://iam-policy-AwsGlueDataBrewSpecificS3BucketPolicy.json
```

搭配 DataBrew 使用加密的 IAM 政策

此AwsGlueDataBrewS3EncryptedPolicy政策會授予必要的許可，以代表非管理使用者存取使用 AWS Key Management Service(AWS KMS) 加密的 S3 物件。

自訂政策，如下所示：

1. 取代政策中的 Amazon S3 路徑，使其指向您要使用的路徑。在範例文字中， *BUCKET-NAME-1/SPECIFIC-OBJECT-NAME*代表特定物件或檔案。 *BUCKET-NAME-2/*代表路徑名稱開頭為 的所有物件 (*)BUCKET-NAME-2/。更新這些項目以命名您正在使用的儲存貯體。
2. （選用） 在 Amazon S3 路徑中使用萬用字元，以進一步限制許可。如需詳細資訊，請參閱 IAM 政策元素：變數和標籤。

   為此，您可以限制動作 s3:PutObject和 的許可s3:PutBucketCORS。只有建立 DataBrew 專案的使用者才需要這些動作，因為這些使用者需要能夠將輸出檔案傳送至 S3。

   如需詳細資訊並查看您可以新增至 Amazon S3 IAM 政策的一些範例，請參閱儲存貯體政策範例。
3. 在 ToUseKms 檔案中尋找下列資源 ARNs。

```
  "arn:aws:kms:AWS-REGION-NAME:AWS-ACCOUNT-ID-WITHOUT-DASHES:key/KEY-IDS",
"arn:aws:kms:AWS-REGION-NAME:AWS-ACCOUNT-ID-WITHOUT-DASHES:key/KEY-IDS"
```

4. 將範例AWS帳戶變更為AWS您的帳戶號碼 （不含連字號）。
5. 將範例清單變更為 ，改為列出您要使用的 IAM 角色。建議您將 IAM 政策範圍調整為盡可能設定的最小許可。不過，您可以允許使用者存取所有 IAM 角色，例如，如果您使用具有範例資料的個人學習帳戶。若要允許清單存取所有 IAM 角色，請將範例清單變更為一個項目："arn:aws:iam::*111122223333*:role/*"。

下表說明此政策授予的許可。

Action	Resource	Description
"s3:GetObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	可讓您預覽檔案。
"s3:ListBucket"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許從專案、資料集和任務列出 Amazon S3 儲存貯體。
"s3:PutObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許將輸出檔案傳送至 S3。
"s3:DeleteObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	允許刪除 DataBrew 建立的物件。
"kms:Decrypt"	"arn:aws:kms:::key/key_ids"	允許解密加密的資料集。
"kms:GenerateDataKey*"	"arn:aws:kms:::key/key_ids"	允許加密任務輸出。

定義 DataBrew 的 AwsGlueDataBrewS3EncryptedPolicy IAM 政策（主控台）

1. 下載 IAM AwsGlueDataBrewS3EncryptedPolicy 政策的 JSON。
2. 登入AWS 管理主控台並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
3. 在導覽窗格中，選擇政策。
4. 針對每個政策，選擇建立政策。
5. 在建立政策畫面上，導覽至 JSON 索引標籤。
6. 將政策 JSON 陳述式貼到編輯器中的範例陳述式上。

7. 確認政策已根據您的 帳戶、安全需求和所需AWS資源進行自訂。如果您需要進行變更，您可以在編輯器中進行變更。
8. 選擇檢閱政策。

定義 DataBrew 的 AwsGlueDataBrewS3EncryptedPolicy IAM 政策 (AWS CLI)

1. 下載適用於 的 JSONAwsGlueDataBrewS3EncryptedPolicy。
2. 依照先前程序的第一個步驟所述自訂政策。
3. 執行下列命令來建立政策。

```
aws iam create-policy --policy-name AwsGlueDataBrewS3EncryptedPolicy --policy-
document file://iam-policy-AwsGlueDataBrewS3EncryptedPolicy.json
```

使用 DataBrew 許可新增使用者或群組

您可以將政策指派給角色，並將角色指派給使用者和群組以管理許可。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 身分（使用者、群組和角色）。

開始之前，您需要至少有一個使用者才能指派許可給 。

使用下列程序為需要在 DataBrew 主控台中工作的使用者設定 DataBrew 許可，或在 CLI 中執行 DataBrew 命令。

設定 DataBrew 許可

1. 為您使用者建立存取金鑰，以使用AWS CLI for DataBrew 和其他開發工具。
2. 啟用AWS 管理主控台存取以允許使用者使用AWS主控台。
3. 為 DataBrew 使用者或群組建立角色。
4. 選擇您正在使用的政策。執行以下任意一項：
   - 如果您建立了 AwsGlueDataBrewCustomUserPolicy，請從清單中選取它。
   - 若要使用 AWS 受管政策，AwsGlueDataBrewFullAccessPolicy請從清單中選擇 。
5. 將該政策指派給角色。
6. 設定角色的信任關係，讓使用者或群組可以擔任相關角色。
   - 如果您不是使用 群組，請信任具有 角色的使用者。

- 如果您使用群組，請信任具有 角色的群組，並將使用者新增至群組。

新增具有資料資源許可的 IAM 角色

您可以使用 IAM 角色來管理一起指派的政策。IAM 角色可由擔任特定角色的人員使用，例如 DataBrew 使用者或 DataBrew 本身。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 角色。

使用下列程序建立 DataBrew 專案存取資料所需的 IAM 角色。

將必要的 IAM 政策連接至 DataBrew 的新 IAM 角色

1. 在導覽窗格中，選擇角色 、建立角色。
2. 針對信任實體的選取類型，選擇卡片標籤AWS服務。
3. 從清單中選擇 DataBrew，然後選擇下一步：許可。
4. **AwsGlueDataBrewDataResourcePolicy** 在搜尋方塊中輸入（您在先前步驟中建立的 IAM 政策）。選取政策，然後選擇下一步：標籤。
5. 選擇下一步：檢閱。
6. 針對角色名稱，輸入 **AwsGlueDataBrewDataAccessRole**，然後選擇建立角色。

設定AWS IAM Identity Center(IAM Identity Center)

使用AWS IAM Identity Center(IAM Identity Center)，您的使用者可以使用簡單的 URL 登入 DataBrew，而無需登入AWS 管理主控台，也不需要AWS帳戶。

設定 IAM Identity Center

1. 開啟 AWS Organizations主控台，如果您還沒有組織，請建立組織。此組織預設會啟用所有功能。

   如需詳細資訊，請參閱AWS IAM Identity Center先決條件和建立和管理組織。
2. 開啟 AWS IAM Identity Center主控台
3. 選擇您的身分來源。

   依預設，您會取得 IAM Identity Center 存放區，以便快速輕鬆地管理使用者。或者，您可以改為連接外部身分提供者，或將AWS Managed Microsoft AD目錄與您的內部部署 Active Directory 連接。在本指南中，我們使用預設的 IAM Identity Center 存放區。

   如需詳細資訊，請參閱AWS IAM Identity Center《 使用者指南》中的選擇您的身分來源。

4. 建立 DataBrew 存取的許可集：

   a. 在 IAM Identity Center 導覽窗格中，選擇AWS帳戶，然後選擇許可集。

   b. 在建立許可集頁面上，選擇建立自訂許可集。

   c. 針對轉送狀態，輸入 `https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`。

      輸入此選項可讓您的使用者直接前往 DataBrew。

   d. 選擇連接AWS受管政策，搜尋 DataBrew，然後選擇 AwsGlueDataBrewFullAccessPolicy。選擇此選項可為您的使用者提供 DataBrew 所需的所有許可。您可以在 中找到更多詳細資訊為主控台使用者新增 IAM 政策。

   e. （選用）選擇建立自訂許可政策，並自訂使用者的許可。

5. 在 IAM Identity Center 導覽窗格中，選擇群組，然後選擇建立群組。輸入群組名稱，然後選擇建立。

6. 將使用者新增至 IAM Identity Center 存放區：

   a. 在 IAM Identity Center 導覽窗格中，選擇使用者。

   b. 在新增使用者畫面上，輸入必要資訊，然後選擇使用密碼設定指示傳送電子郵件給使用者。使用者應會收到有關下一個設定步驟的電子郵件。

   c. 選擇下一步：群組，選擇您想要的群組，然後選擇新增使用者。

      使用者應會收到邀請他們使用 SSO 的電子郵件。在此電子郵件中，他們需要選擇接受邀請並設定密碼。他們也可以在電子郵件中找到入口網站 URL。他們可以使用此 URL 來存取 DataBrew。

7. 將每個使用者指派給 帳戶：

   a. 開啟 IAM Identity Center 主控台，然後在導覽窗格中選擇AWS帳戶。

   b. 選擇AWS組織，然後選擇AWS帳戶。

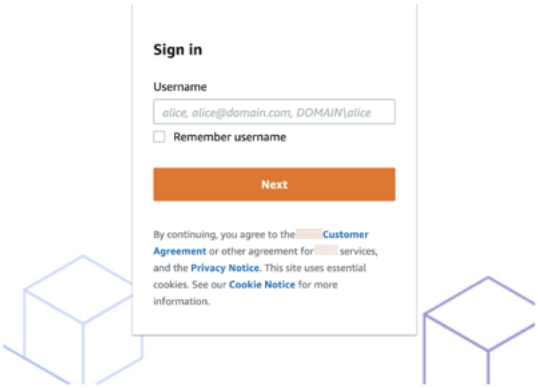
   c. 在指派使用者畫面上，選擇群組索引標籤，然後選擇您想要的群組。

   d. 選擇 Next: Permission sets (下一步：許可集合)。

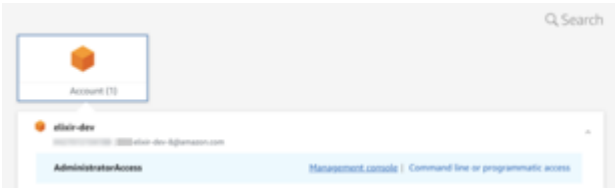
   e. 選擇 DataBrew 的許可集，然後選擇完成。

啟用 IAM Identity Center 的使用者登入步驟

1. AWS使用啟用 IAM Identity Center 的帳戶登入。



2. 按一下AWS帳戶身分



3. 按一下 管理主控台，即可將一鍵式重新導向至 DataBrew 主控台。

在 JupyterLab 中使用 DataBrew 作為延伸模組

> **Warning**
>
> AWS Glue DataBrew JupyterLab 延伸支援將於 2024 年 12 月 31 日結束，因為 JupyterLab 3 將終止支援。如需詳細資訊，請參閱維護結束 JupyterLab 3。

如果您偏好在 Jupyter 筆記本環境中準備資料，則可以在 JupyterLab AWS Glue DataBrew中使用 的所有功能。

JupyterLab 是 Jupyter Notebook 的 Web 型互動式開發環境。在本機 JupyterLab 網頁中，您可以為終端機、SQL 工作階段、Python 等新增區段。安裝AWS Glue DataBrew擴充功能後，您可以為 DataBrew 主控台新增區段。它會直接從 JupyterLab 環境執行任何現有的筆記本或其他擴充功能。

主題

- 先決條件
- 設定 JupyterLab 以使用延伸模組
- 啟用 JupyterLab 的 DataBrew 延伸模組

先決條件

開始之前，請設定下列項目：

- AWS帳戶 – 如果您還沒有帳戶，請從 開始設定新AWS帳戶。
- 可存取 DataBrew 所需許可的AWS Identity and Access Management(IAM) 使用者 – 如需詳細資訊，請參閱 使用 DataBrew 許可新增使用者或群組。
- 要在 DataBrew 操作中使用的 IAM 角色 – 如果`AwsGlueDataBrewDataAccessRole`已設定 ，您可以使用預設值。若要設定其他 IAM 角色，請參閱 新增具有資料資源許可的 IAM 角色。
- JupyterLab 安裝 (2.2.6 版或更新版本） – 如需詳細資訊，請參閱 JupyterLab 文件中的下列主題：
  - JupyterLab 先決條件
  - JupyterLab 安裝 – 建議使用 `pip install jupyterlab`。
- Node.js 安裝 (12.0 版或更新版本）。
- An AWS Command Line Interface(AWS CLI) 安裝 – 如需詳細資訊，請參閱 設定AWS CLI。
- Jupyter AWS代理安裝 (`pip install aws-jupyter-proxy`) – 此延伸項目會與服務AWS端點搭配使用，以安全地傳遞您的AWS登入資料。如需詳細資訊，請參閱 GitHub 上的 aws-jupyter-proxy。

若要驗證是否已安裝先決條件，您可以在命令列執行類似下列的測試，如下列範例所示。

```
echo "
AWS CLI:"
which aws
aws --version
aws configure list
aws sts get-caller-identity

echo "
Python (current environment):"
which python
python --version

echo "
Node.JS:"
which node
node --version

echo "
```

```
Jupyter:"
where jupyter
jupyter --version
jupyter serverextension list
pip3 freeze | grep jupyter
```

輸出看起來應該如下。目錄因作業系統和組態而異。

```
AWS CLI:
/usr/local/bin/aws
aws-cli/2.1.2 Python/3.7.4 Darwin/19.6.0 exe/x86_64
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************VXW4 shared-credentials-file
secret_key     ****************MRJN shared-credentials-file
    region                us-east-1      config-file    ~/.aws/config
{
    "UserId": "",
    "Account": "111122223333",
    "Arn": "arn:aws:iam::111122223333:user/user2"
}

Python (current environment):
/usr/local/opt/python /libexec/bin/python
Python 3.8.5

Node.JS:
/usr/local/bin/node
v15.0.1

Jupyter:
/usr/local/bin/jupyter
jupyter core     : 4.6.3
jupyter-notebook : 6.0.3
qtconsole        : 4.7.5
ipython          : 7.16.1
ipykernel        : 5.3.2
jupyter client   : 6.1.6
jupyter lab      : 2.2.9
nbconvert        : 5.6.1
ipywidgets       : 7.5.1
nbformat         : 5.0.7
```

```
traitlets        : 4.3.3

config dir: /usr/local/etc/jupyter
    aws_jupyter_proxy  enabled
    - Validating...
      aws_jupyter_proxy  OK
    jupyterlab  enabled
    - Validating...
      jupyterlab 2.2.9 OK

aws-jupyter-proxy==0.1.0
jupyter-client==6.1.7
jupyter-core==4.7.0
jupyterlab==2.2.9
jupyterlab-pygments==0.1.2
jupyterlab-server==1.2.0
```

設定 JupyterLab 以使用延伸模組

安裝 JupyterLab 後，您需要將其設定為保護資料存取和啟用伺服器延伸。

設定密碼和加密

1. 設定密碼來保護您計劃在擴充功能中新增的資料。Jupyter 會提供密碼公用程式。執行以下命令，並在系統提示時輸入您慣用的密碼。

   ```
   jupyter notebook password
   ```

   輸出類似如下所示。

   ```
   Enter password:
   Verify password:
   [NotebookPasswordApp] Wrote hashed password to /home/ubuntu/.jupyter/
   jupyter_notebook_config.json
   ```

2. 在 Jupyter 伺服器上啟用加密。如果您在本機電腦上安裝 Jupyter，且沒有人可以透過網路存取，則可以略過此步驟。

   若要使用 Transport Layer Security (TLS) 設定加密，請建立為您的環境自訂的憑證。如需詳細資訊，請在 Jupyter 文件中使用 Let's Encrypt 來保護伺服器。

3. 若要啟動 JupyterLab，請在命令提示字元中執行下列命令。

```
jupyter lab
```

如需詳細資訊，請參閱 JupyterLab 文件中的啟動 JupyterLab。

4. 當 JupyterLab 正在執行時，您可以在類似如下的 URL 中存取它：http://localhost:8888/lab。如果您設定加密，請使用 `https`而非 `http`。如果您自訂連接埠，請取代連接埠號碼，而不是 8888。

使用下列程序來啟用第三方延伸模組。

在 JupyterLab 中啟用第三方延伸模組

1. 在 JupyterLab 網頁上，選擇左側選單中的延伸管理員圖示。
2. 閱讀有關執行第三方延伸模組風險的警告。僅從您信任的開發人員安裝擴充功能。
3. 若要在 JupyterLab 中啟用第三方延伸模組，請選擇啟用。
4. 依照提示重建和重新載入 JupyterLab。

啟用 JupyterLab 的 DataBrew 延伸模組

在啟用擴充功能的情況下安全地安裝 JupyterLab 之後，請安裝 DataBrew 擴充功能，以便在筆記本中執行 DataBrew。

安裝 DataBrew 的擴充功能 （主控台）

1. 若要啟動 JupyterLab，請在命令提示字元中執行下列命令。

```
jupyter lab
```

2. 在 JupyterLab 網頁上，選擇左側選單中的延伸管理員圖示。
3. 為左上角的搜尋輸入「**`brew`**」來搜尋 DataBrew 延伸模組。
4. 在清單中找到 aws_glue_databrew_jupyter，但不要按一下它。如果您按一下副檔名的反白名稱，則會開啟新的瀏覽器視窗，其中包含 GitHub 上的 aws_glue_databrew_jupyter 頁面。
5. 若要安裝 DataBrew 延伸模組，請選擇下列其中一項：

   - 在命令列中，執行 `jupyter labextension install aws_glue_databrew_jupyter`。
   - 在「aws_glue_databrew_jupyter」下方的灰色字母下，選擇擴充卡底部的安裝。

DataBrew 延伸模組與 JupyterLab 1.2 版和 2.x 版相容。

6. 若要驗證是否已安裝，請執行 `jupyter labextension list`。輸出看起來應該如下。

```
JupyterLab v2.2.9
Known labextensions:
   app dir: /usr/local/share/jupyter/lab  # varies by OS
        aws_glue_databrew_jupyter v1.0.1  enabled  OK
```

7. 使用下列其中一項重建 JupyterLab：

   - 在命令提示字元中，執行 `jupyter lab build`。
   - 在網頁中，選擇左上角的重建。

8. 當組建完成時，請執行下列其中一項操作：

   - 在命令提示字元中，執行 `jupyter lab`。
   - 在網頁中，選擇建置完成訊息上的重新載入。

9. 在 JupyterLab 網頁中，選擇左側選單中的圖示來關閉延伸管理員。

   若要開啟延伸模組，請從啟動AWS Glue DataBrew器索引標籤上的其他區段選擇啟動。 延伸模組會使用您目前的AWS CLI組態進行存取金鑰和AWS區域設定。

完成設定後，您可以使用 AWS Glue DataBrew標籤，從 JupyterLab 內與 DataBrew 互動。

入門AWS Glue DataBrew

您可以使用下列教學課程來引導您建立第一個 DataBrew 專案。您可以載入範例資料集、在該資料集上執行轉換、建置配方來擷取這些轉換，以及執行任務以將轉換後的資料寫入 Amazon S3。

主題

先決條件

在繼續之前，請遵循 中的適用說明設定AWS Glue DataBrew。然後繼續 步驟 1：建立專案。

步驟 1：建立專案

在此步驟中，您可以使用 DataBrew 主控台快速開始使用範例專案。

建立專案

1. 登入AWS 管理主控台，並在 https：//https://console.aws.amazon.com/databrew/ 開啟 DataBrew 主控台。
2. 請確定您的AWS區域已在 DataBrew 主控台的右上角選取。如需 DataBrew 支援AWS的區域清單，請參閱《》中的 DataBrew 端點和配額AWS 一般參考。
3. 在導覽窗格中，選擇專案，然後選擇建立專案。
4. 在專案詳細資訊窗格中，執行下列動作：
   - 針對專案名稱，輸入 `chess-project`。

- 針對連接的配方，建立新的配方。提供配方的建議名稱 (chess-project-recipe)。

5. 在選取資料集窗格中，選擇範例檔案。
6. 在範例檔案窗格中，選擇知名的西洋棋遊戲移動。此資料集包含超過 20，000 個西洋棋遊戲的詳細資訊。

   對於資料集名稱，會提供資料集的建議名稱 (chess-games)。
7. 在存取許可窗格中，選擇 AwsGlueDataBrewDataAccessRole。這是一個服務連結角色，可讓 DataBrew 代表您存取您的 Amazon S3 儲存貯體。
8. 選擇建立專案，然後等待 DataBrew 完成準備專案。視窗看起來類似以下內容。

   您看到的資料代表來自chess-games資料集的範例。根據預設，範例由資料集的前 500 個資料列組成。您可以稍後變更此專案設定。

   工具列可讓您存取數百個可套用至資料的資料轉換。

   DataBrew 主控台右側的配方窗格會追蹤您到目前為止套用的轉換。

步驟 2：摘要資料

在此步驟中，您會建置 DataBrew 配方，這是一組可套用至此資料集和其他類似資料集的轉換。當配方完成時，您會發佈它，使其可供使用。

在西洋棋遊戲中，玩家可以根據與其他玩家相比的效能進行評分。(如需詳細資訊，請參閱 https://en.wikipedia.org/wiki/Chess_rating_system)。在本教學課程中，您只專注於兩個玩家都是 A 級的遊戲，這表示他們的評分為 1800 或更高。

摘要資料

1. 在轉換工具列上，選擇篩選、依條件、大於或等於。
2. 設定這些選項，如下所示：

   - 來源資料欄 - white_rating
   - 篩選條件 – 大於或等於 1800

   若要查看轉換的運作方式，請選擇預覽變更。接著選擇 Apply (套用)。
3. 重複上一個步驟，但這次將來源資料欄設定為 black_rating。套用變更後，範例資料只會包含兩側玩家 （ 黑色和白色 ） 為 A 級或更高等級的遊戲。

4. 摘要資料，以判斷每一端獲得多少遊戲。若要這樣做，請在轉換工具列上，選擇群組。
5. 對於群組屬性，請執行下列動作：

   a. 在第一列中，選擇 winner做為資料欄名稱。將彙總設定為分組依據。

   b. 在第二列中，選擇 victory_status做為資料欄名稱。將彙總設定為分組依據。

   c. 選擇新增另一個資料欄。

   d. 在第三列中，選擇winner資料欄名稱。將彙總設定為計數。

   e. 針對群組類型，選擇群組做為新資料表。預覽窗格會顯示結果的外觀。

   f. 選擇完成。
6. 在配方窗格中，選擇發佈以儲存您的工作。
7. 在版本描述中，輸入我的配方的第一個版本。然後選擇發佈。

步驟 3：新增更多轉換

在此步驟中，您會將更多轉換新增至配方，並發佈另一個版本的配方。為了精簡我們的範例，我們使用並非所有西洋棋遊戲都會產生明確的獲勝者的資訊；有些遊戲會進行繪製。

新增更多配方轉換並重新發佈

1. 從轉換工具列中，選擇篩選、依條件、不是移除播放至繪製的遊戲。
2. 設定這些選項，如下所示：

   - 來源資料欄 - victory_status
   - 篩選條件 – 不是 draw

   若要將此轉換新增至您的配方，請選擇套用。
3. 變更 中的資料，victory_status使其更有意義。若要這樣做，請從轉換工具列選擇清除、取代、取代值或模式。
4. 設定這些選項，如下所示：

   - 來源資料欄 - victory_status
   - 指定要取代的值 – 值或模式
   - 要取代的值 - mate
   - 以值取代 - checkmate

若要將此轉換新增至您的配方，請選擇套用。

5. 重複上一個步驟，但resign變更為 other player resigned。
6. 重複上一個步驟，但outoftime變更為 time ran out。
7. 在配方窗格中，選擇發佈以儲存您的工作。

步驟 4：檢閱 DataBrew 資源

現在您已使用範例專案，請檢閱您目前建立的 DataBrew 資源。

若要檢閱 DataBrew 資源

1. 在導覽窗格中，選擇資料集。

   當您建立範例專案時，DataBrew 會為您建立資料集 (chess-games)。來源資料檔案存放在 Amazon S3 中，且為 Microsoft Excel 格式 (chess-games.xlsx)。該檔案包含來自超過 20，000 款西洋棋遊戲的中繼資料。chess-games 資料集提供 DataBrew 讀取該檔案中資料所需的資訊。
2. 在導覽窗格中，選擇專案。

   您應該會在先前的步驟 () 中看到您使用的專案chess-project。每個專案都需要資料集，在此情況下為 chess-games。每個專案也都需要配方，因此您可以在進行時新增資料轉換步驟。當您建立此範例專案時，DataBrew 會為您建立新的 （空白） 配方，並將其連接至專案。
3. 在導覽窗格中，選擇配方，然後在配方名稱欄中，選擇 chess-project-recipe。這會顯示 DataBrew 為專案建立的配方，以及您在專案中新增轉換步驟所精簡的配方。
4. 在左側，檢視已發佈的配方版本。選擇其中一項以檢視其配方步驟索引標籤，其中顯示該版本的配方詳細資訊和步驟。
5. 檢視資料歷程索引標籤，其中顯示資料的來源和使用方式。如需詳細資訊，請選擇圖表中的任何圖示。

步驟 5：建立資料設定檔

當您使用 處理專案時，DataBrew 會顯示統計資料，例如範例中的資料列數，以及每個資料欄中唯一值的分佈。這些統計資料等都代表範例的描述檔。

若要請求資料設定檔，請建立並執行設定檔任務。

描述資料集

1. 在導覽窗格中，選擇任務。
2. 在設定檔任務索引標籤上，選擇建立任務。
3. 針對任務名稱，輸入 chess-data-profile。
4. 針對任務類型，選擇建立設定檔任務。
5. 在任務輸入窗格中，執行下列動作：
   - 針對執行，選擇資料集。
   - 選擇選取資料集以檢視可用資料集的清單，然後選擇 chess-games。
6. 在任務輸出設定窗格中，執行下列動作：
   - 針對檔案類型，選擇 JSON (JavaScript 物件標記法）。
   - 選擇 S3 位置以檢視可用的 Amazon S3 儲存貯體清單，然後選擇要使用的儲存貯體。然後選擇瀏覽。在資料夾清單中，選擇 databrew-output，然後選擇選取。
7. 在存取許可窗格中，選擇 AwsGlueDataBrewDataAccessRole。這是服務連結角色，可讓 DataBrew 代表您存取您的 Amazon S3 儲存貯體。
8. 選擇建立和執行任務。DataBrew 會使用您的設定建立任務，然後執行該任務。
9. 在任務執行歷史記錄窗格中，等待任務狀態從 變更為 Running Succeeded。
10. 若要檢視設定檔，請選擇檢視設定檔：

隨即顯示 DATASETS 視窗。花一些時間探索下列索引標籤：

- 資料集預覽
- 設定檔概觀
- 資料欄統計資料
- 資料歷程統計資料

步驟 6：轉換資料集

到目前為止，您只對資料集的範例測試配方。現在是時候透過建立 DataBrew 配方任務來轉換整個資料集。

當任務執行時，DataBrew 會將您的配方套用至資料集中的所有資料，並將轉換的資料寫入 Amazon S3 儲存貯體。轉換的資料與原始資料集分開。DataBrew 不會變更來源資料。

在繼續之前，請確定您的帳戶中有可寫入的 Amazon S3 儲存貯體。在該儲存貯體中，建立資料夾以從 DataBrew 擷取任務輸出。若要執行這些步驟，請使用下列程序。

建立 S3 儲存貯體和資料夾以擷取任務輸出

1. 登入AWS 管理主控台並開啟位於 https：//https://console.aws.amazon.com/databrew/ 的 Amazon S3 主控台。

   如果您已經有可用的 Amazon S3 儲存貯體，且擁有其寫入許可，請略過下一個步驟。
2. 如果您沒有 Amazon S3 儲存貯體，請選擇建立儲存貯體。針對儲存貯體名稱，輸入新儲存貯體的唯一名稱。選擇建立儲存貯體。
3. 從儲存貯體清單中，選擇您要使用的儲存貯體。
4. 選擇 Create folder (建立資料夾)。
5. 針對資料夾名稱，輸入 `databrew-output`，然後選擇建立資料夾。

建立 Amazon S3 儲存貯體和資料夾以包含任務後，請使用下列程序執行任務。

建立和執行配方任務

1. 在導覽窗格中，選擇任務。
2. 在配方任務索引標籤上，選擇建立任務。
3. 針對任務名稱，輸入 `chess-winner-summary`。
4. 針對任務類型，選擇建立配方任務。
5. 在任務輸入窗格中，執行下列動作：
   - 針對執行時，選擇資料集。
   - 選擇選取資料集以檢視可用資料集的清單，然後選擇 `chess-games`。
   - 選擇選取配方以檢視可用配方的清單，然後選擇 `chess-project-recipe`。
6. 在任務輸出設定窗格中，執行下列動作：
   - 檔案類型 – 選擇 CSV （逗號分隔值）。
   - S3 位置 - 選擇此欄位以檢視可用的 Amazon S3 儲存貯體清單，然後選擇要使用的儲存貯體。然後選擇瀏覽。在資料夾清單中，選擇 `databrew-output`，然後選擇選取。

7. 在存取許可窗格中，選擇 AwsGlueDataBrewDataAccessRole。此服務連結角色可讓 DataBrew 代表您存取您的 Amazon S3 儲存貯體。
8. 選擇建立和執行任務。DataBrew 會使用您的設定建立任務，然後執行該任務。
9. 在任務執行歷史記錄窗格中，等待任務狀態從 變更為 Running Succeeded。
10. 選擇輸出以存取 Amazon S3 主控台。選擇您的 S3 儲存貯體 databrew-output，然後選擇資料夾以存取任務輸出。
11. （選用）選擇下載以下載檔案並檢視其內容。

步驟 7：（選用）清除

演練已完成。您可以繼續使用您建立的 DataBrew 和 Amazon S3 資源，或刪除它們。

清理資源

1. 在 https：//https://console.aws.amazon.com/databrew/ 開啟 DataBrew 主控台，然後在導覽窗格中選擇專案。
2. 選擇您的專案（範例專案)。對於 Actions (動作)，請選擇 Delete (刪除)。
3. 在刪除範例專案窗格中，選擇刪除連接的配方。然後選擇刪除。您的專案及其配方和任務將被刪除。
4. 在導覽窗格中，選擇資料集。
5. 選擇您的資料集 (chess-games)，然後針對動作，選擇刪除。
6. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。刪除 databrew-output 資料夾及其內容。

   （選用）如果您確定不再需要 Amazon S3 儲存貯體，可以將其刪除。

使用 連線至資料AWS Glue DataBrew

在 中AWS Glue DataBrew，資料集代表從檔案上傳或存放在其他位置的資料。例如，資料可以存放在 Amazon S3、支援的 JDBC 資料來源或AWS Glue Data Catalog 中。如果您不直接將檔案上傳至 DataBrew，資料集也會包含 DataBrew 如何連線至資料的詳細資訊。

當您建立資料集 （ 例如 `inventory-dataset`) 時，您只需輸入連線詳細資訊一次。此時，DataBrew 可以為您存取基礎資料。透過此方法，您可以建立專案並開發資料的轉換，而不必擔心連線詳細資訊或檔案格式。

主題

資料來源支援的檔案類型

下列檔案需求適用於存放在 Amazon S3 中的檔案，以及您從本機磁碟機上傳的檔案。DataBrew 支援下列檔案格式：逗號分隔值 (CSV)、Microsoft Excel、JSON、ORC 和 Parquet。如果檔案是其中一個支援的類型，則可以使用具有非標準副檔名或無副檔名的檔案。

如果 DataBrew 無法推斷檔案類型，請務必自行選取正確的檔案類型 (CSV、Excel、JSON、ORC 或 Parquet)。支援壓縮的 CSV、JSON、ORC 和 Parquet 檔案，但 CSV 和 JSON 檔案必須包含壓縮轉碼器作為副檔名。如果您要匯入資料夾，則資料夾中的所有檔案都必須是相同的檔案類型。

檔案格式和支援的壓縮演算法會顯示在下表中。

> **Note**
>
> CSV、Excel 和 JSON 檔案必須以 Unicode (UTF-8) 編碼。

Format (格式)	副檔名 （選用）	壓縮檔案的延伸模組 （必要）

逗號分隔值	`.csv`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Microsoft Excel 工作手冊	`.xlsx`	無壓縮支援

JSON (JSON 文件和 JSON 行）	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Apache ORC	`.orc`	`.zlib` `.snappy`

Apache Parquet	`.parquet`	`.gz`
		`.snappy`
		`.lz4`

資料來源和輸出支援的連線

您可以連線至 DataBrew 配方任務的下列資料來源。其中包括不是您直接上傳至 DataBrew 之檔案的任何資料來源。您使用的資料來源可能稱為資料庫、資料倉儲或其他項目。我們將所有資料提供者稱為資料來源或連線。

您可以使用下列任何一項做為資料來源來建立資料集。

您也可以使用透過 Amazon RDS 支援的 Amazon S3 AWS Glue Data Catalog或 JDBC 資料庫來輸出 DataBrew 配方任務。Amazon AppFlow 和AWS Data Exchange不支援 DataBrew 配方任務輸出的資料存放區。

- Amazon S3

  您可以使用 S3 來存放和保護任意數量的資料。若要建立資料集，您可以指定 S3 URL，讓 DataBrew 可以存取資料檔案，例如： `s3://your-bucket-name/inventory-data.csv`

  DataBrew 也可以讀取 S3 資料夾中的所有檔案，這表示您可以建立跨越多個檔案的資料集。若要這樣做，請以此格式指定 S3 URL：`s3://your-bucket-name/your-folder-name/`。

  DataBrew 僅支援下列 Amazon S3 儲存類別：標準、降低備援、標準 – IA 和 S3 單區域 – IA。DataBrew 會忽略其他儲存類別的檔案。DataBrew 也會忽略空白檔案 （包含 0 個位元組的檔案）。如需 Amazon S3 儲存類別的詳細資訊，請參閱《Amazon S3 主控台使用者指南》中的使用 Amazon S3 儲存類別。 Amazon S3

- AWS Glue Data Catalog

  您可以使用 Data Catalog 來定義對儲存在AWS雲端中資料的參考。使用 Data Catalog，您可以在下列服務中建立個別資料表的連線：

  - Data Catalog Amazon S3
  - Data Catalog Amazon Redshift
  - Data Catalog Amazon RDS
  - AWS Glue

  DataBrew 也可以讀取 Amazon S3 資料夾中的所有檔案，這表示您可以建立跨越多個檔案的資料集。若要這樣做，請以此格式指定 Amazon S3 URL： `s3://your-bucket-name/your-folder-name/`

  若要與 DataBrew 搭配使用， 中定義的 Amazon S3 資料表AWS Glue Data Catalog必須新增名為 的資料表屬性，`classification`該屬性會將資料格式識別為 `csv`、 `json`或 `parquet`，並將 `typeOfData`識別為 `file`。如果在建立資料表時未新增資料表屬性，您可以使用AWS Glue主控台新增資料表屬性。

  DataBrew 僅支援 Amazon S3 儲存類別標準、降低備援、標準 – IA 和 S3 單區域 – IA。DataBrew 會忽略其他儲存類別的檔案。DataBrew 也會忽略空白檔案 （包含 0 個位元組的檔案）。如需

Amazon S3 儲存類別的詳細資訊，請參閱《Amazon S3 主控台使用者指南》中的使用 Amazon S3 儲存類別。 Amazon S3

如果建立適當的資源政策，DataBrew 也可以從其他帳戶存取AWS Glue Data Catalog S3 資料表。您可以在主控台的AWS GlueData Catalog 下的設定索引標籤上建立政策。以下是專門針對單一 的範例政策AWS 區域。

> **Warning**
>
> 這是高度寬鬆的資源政策，授予 的資料目錄*$ACCOUNT_TO*不受限制的存取權*$ACCOUNT_FROM*。在大多數情況下，我們建議您將資源政策鎖定在特定目錄或資料表。如需詳細資訊，請參閱《 AWS Glue開發人員指南》中的AWS Glue存取控制的資源政策。

在某些情況下，您可能想要在AWS Glue DataBrew中使用 *$ACCOUNT_TO*AWS Glue Data Catalog S3 資料表在 中建立專案或執行任務*$ACCOUNT_FROM*，該資料表指向也在 中的 S3 位置*$ACCOUNT_FROM*。在這種情況下，在 中建立專案和任務時使用的 IAM 角色*$ACCOUNT_TO*必須具有從 列出和取得該 S3 位置中物件的許可*$ACCOUNT_FROM*。如需詳細資訊，請參閱《 AWS Glue開發人員指南》中的授予跨帳戶存取權。

- 使用 JDBC 驅動程式連線的資料

  您可以使用支援的 JDBC 驅動程式連線至資料，以建立資料集。如需詳細資訊，請參閱搭配 使用驅動程式AWS Glue DataBrew。

  DataBrew 使用 Java Database Connectivity (JDBC) 正式支援下列資料來源：

  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Amazon Redshift
  - 適用於 Spark 的 Snowflake 連接器

  資料來源可以位於您可以從 DataBrew 與其連線的任何位置。此清單僅包含我們已測試且因此可支援的 JDBC 連線。

  Amazon Redshift 和 Snowflake Connector for Spark 資料來源可以透過下列其中一種方式連接：

  - 使用資料表名稱。

- 使用跨越多個資料表和操作的 SQL 查詢。

當您啟動專案或任務執行時，會執行 SQL 查詢。

若要連線到需要未列出 JDBC 驅動程式的資料，請確定驅動程式與 JDK 8 相容。若要使用驅動程式，請將它存放在 S3 中的儲存貯體中，您可以在其中使用 DataBrew 的 IAM 角色存取它。然後將資料集指向驅動程式檔案。如需詳細資訊，請參閱搭配 使用驅動程式AWS Glue DataBrew。

SQL 型資料集的範例查詢：

```
SELECT
    *
FROM
    public.customer as c
JOIN
    public.customer_address as ca on c.current_address=ca.current_address
WHERE
    ca.address_id>0 AND ca.address_id<10001 ORDER BY ca.address_id
```

自訂 SQL 的限制

如果您使用 JDBC 連線來存取 DataBrew 資料集的資料，請記住下列事項：

- AWS Glue DataBrew不會驗證您在建立資料集時提供的自訂 SQL。當您啟動專案或任務執行時，將會執行 SQL 查詢。DataBrew 會接受您提供的查詢，並使用預設或提供的 JDBC 驅動程式將其傳遞至資料庫引擎。
- 在專案或任務中使用無效查詢時，建立的資料集將會失敗。在建立資料集之前驗證您的查詢。
- 驗證 SQL 功能僅適用於以 Amazon Redshift 為基礎的資料來源。
- 如果您想要在專案中使用資料集，請將 SQL 查詢執行時間限制在三分鐘以下，以避免在專案載入期間逾時。建立專案之前，請檢查查詢執行時間。

- Amazon AppFlow

使用 Amazon AppFlow，您可以從第三方Software-as-a-Service (SaaS) 應用程式將資料傳輸到 Amazon S3，例如 Salesforce、Zendesk、Slack 和 ServiceNow。然後，您可以使用資料來建立 DataBrew 資料集。

在 Amazon AppFlow 中，您可以建立連線和流程，以在第三方應用程式和目的地應用程式之間傳輸資料。搭配使用 Amazon AppFlow 與 DataBrew 時，請確定 Amazon AppFlow 目的地應用程式是 Amazon S3。Amazon S3 以外的 Amazon AppFlow 目的地應用程式不會出現在 DataBrew 主控

台中。如需從第三方應用程式傳輸資料和建立 Amazon AppFlow 連線和流程的詳細資訊，請參閱 Amazon AppFlow 文件。

當您在 DataBrew 的資料集索引標籤中選擇連接新資料集，然後按一下 Amazon AppFlow 時，您會看到 Amazon AppFlow 中設定 Amazon S3 作為目的地應用程式的所有流程。若要將流程的資料用於資料集，請選擇該流程。

在 DataBrew 主控台中選擇建立流程、管理流程和檢視 Amazon AppFlow 的詳細資訊，會開啟 Amazon AppFlow 主控台，讓您可以執行這些任務。 AppFlow

從 Amazon AppFlow 建立資料集之後，您可以執行流程，並在檢視資料集詳細資訊或任務詳細資訊時檢視最新的流程執行詳細資訊。當您在 DataBrew 中執行流程時，資料集會在 S3 中更新，並準備好在 DataBrew 中使用。

當您在 DataBrew 主控台中選取 Amazon AppFlow 流程來建立資料集時，可能會發生下列情況：

- 資料尚未彙總 - 如果流程觸發條件是隨需執行或按排程執行並搭配完整資料傳輸，請務必先彙總流程的資料，再使用它來建立 DataBrew 資料集。彙總流程會將流程中的所有記錄合併為單一檔案。具有觸發類型的流程 使用增量資料傳輸按排程執行，或在事件上執行不需要彙總。若要彙總 Amazon AppFlow 中的資料，請選擇編輯流程組態 > 目的地詳細資訊 > 其他設定 > 資料傳輸偏好設定。
- 流程尚未執行 - 如果流程的執行狀態為空，則表示下列其中一項：
  - 如果執行流程的觸發是隨需執行，表示流程尚未執行。
  - 如果執行流程的觸發是在事件上執行，則觸發事件尚未發生。
  - 如果執行流程的觸發條件是按排程執行，則尚未發生排定的執行。

  使用流程建立資料集之前，請選擇該流程的執行流程。

  如需詳細資訊，請參閱《Amazon AppFlow 使用者指南》中的 Amazon AppFlow 流程。 AppFlow
- AWS Data Exchange

  您可以從數百個可用的第三方資料來源中進行選擇AWS Data Exchange。透過訂閱這些資料來源，您可以取得最新版本up-to-date的資料。

  若要建立資料集，您可以指定您訂閱並有權使用AWS Data Exchange的資料產品名稱。

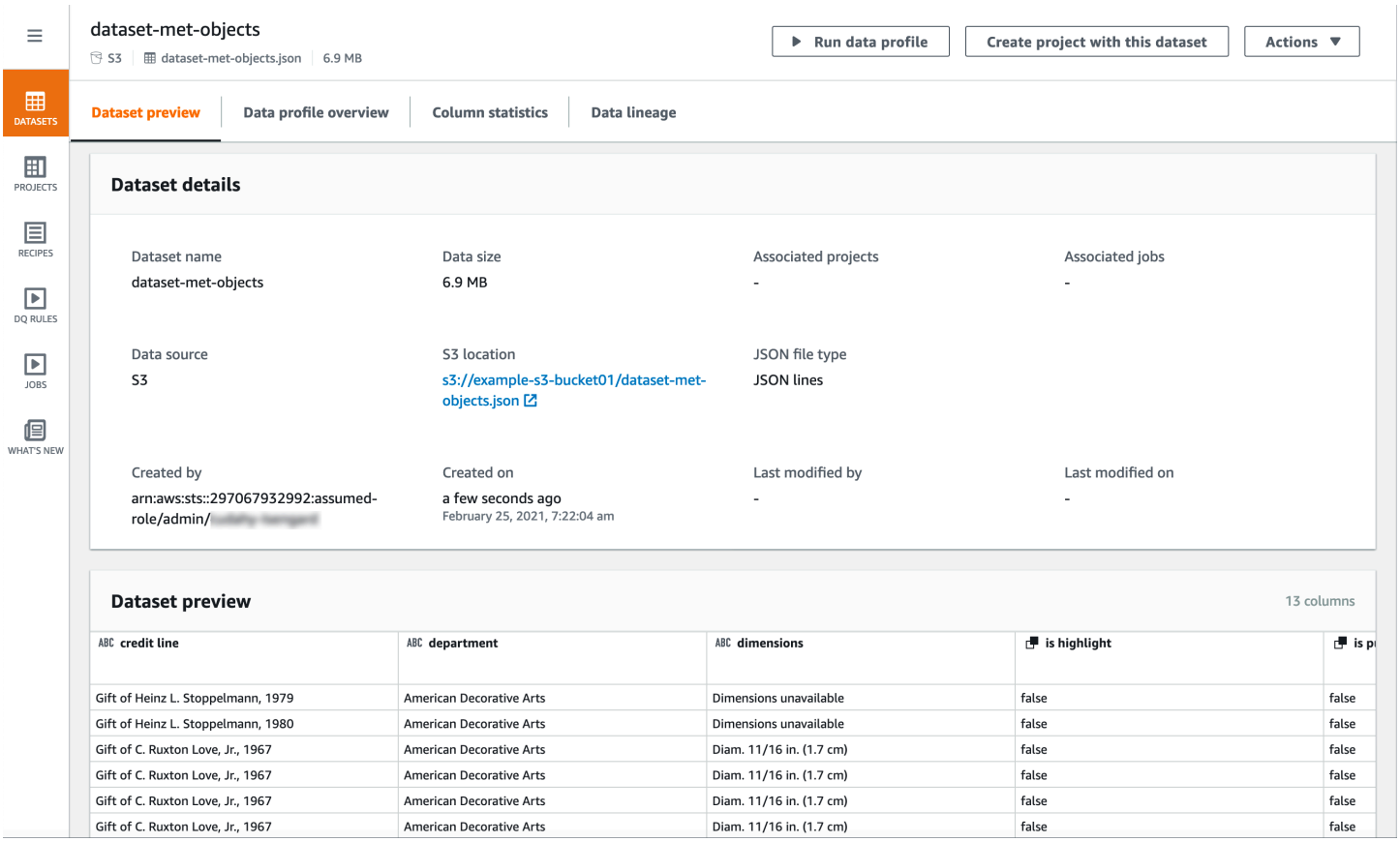
在 中使用資料集AWS Glue DataBrew

若要在 DataBrew 主控台中檢視資料集的清單，請選擇左側的資料集。在資料集頁面中，您可以按一下每個資料集的名稱，或從其內容功能表中選擇動作、編輯，以檢視其詳細資訊。

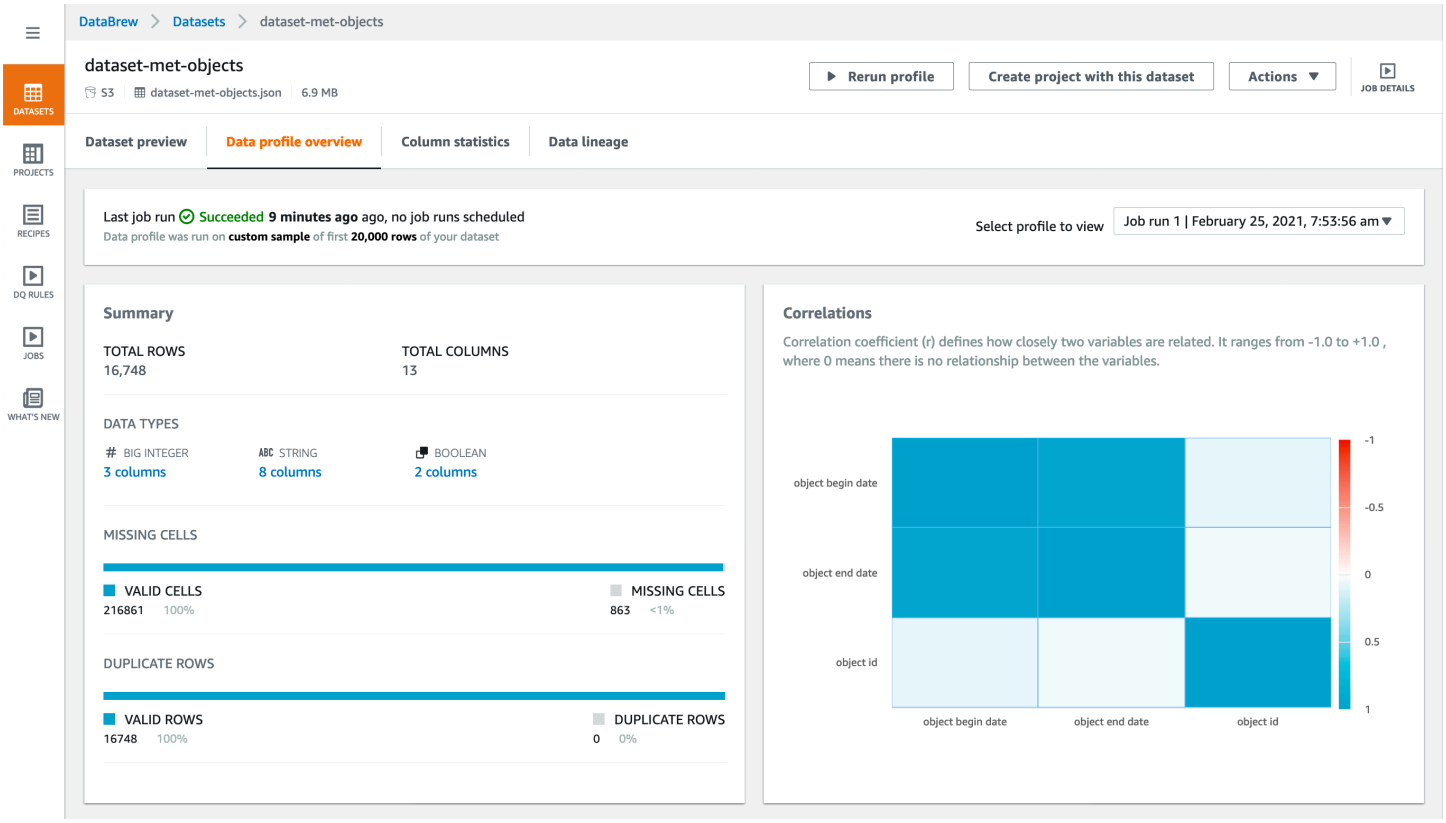
若要建立新的資料集，請選擇資料集、連接新的資料集。不同的資料來源有不同的連線參數，您可以輸入這些參數，以便 DataBrew 可以連線。當您儲存連線並選擇建立資料集時，DataBrew 會連線至您的資料並開始載入資料。如需詳細資訊，請參閱連線至您的資料。

資料集頁面具有下列元素，可協助您探索資料。

資料集預覽 – 在此索引標籤上，您可以找到資料集的連線資訊，以及資料集整體結構的概觀，如下所示。

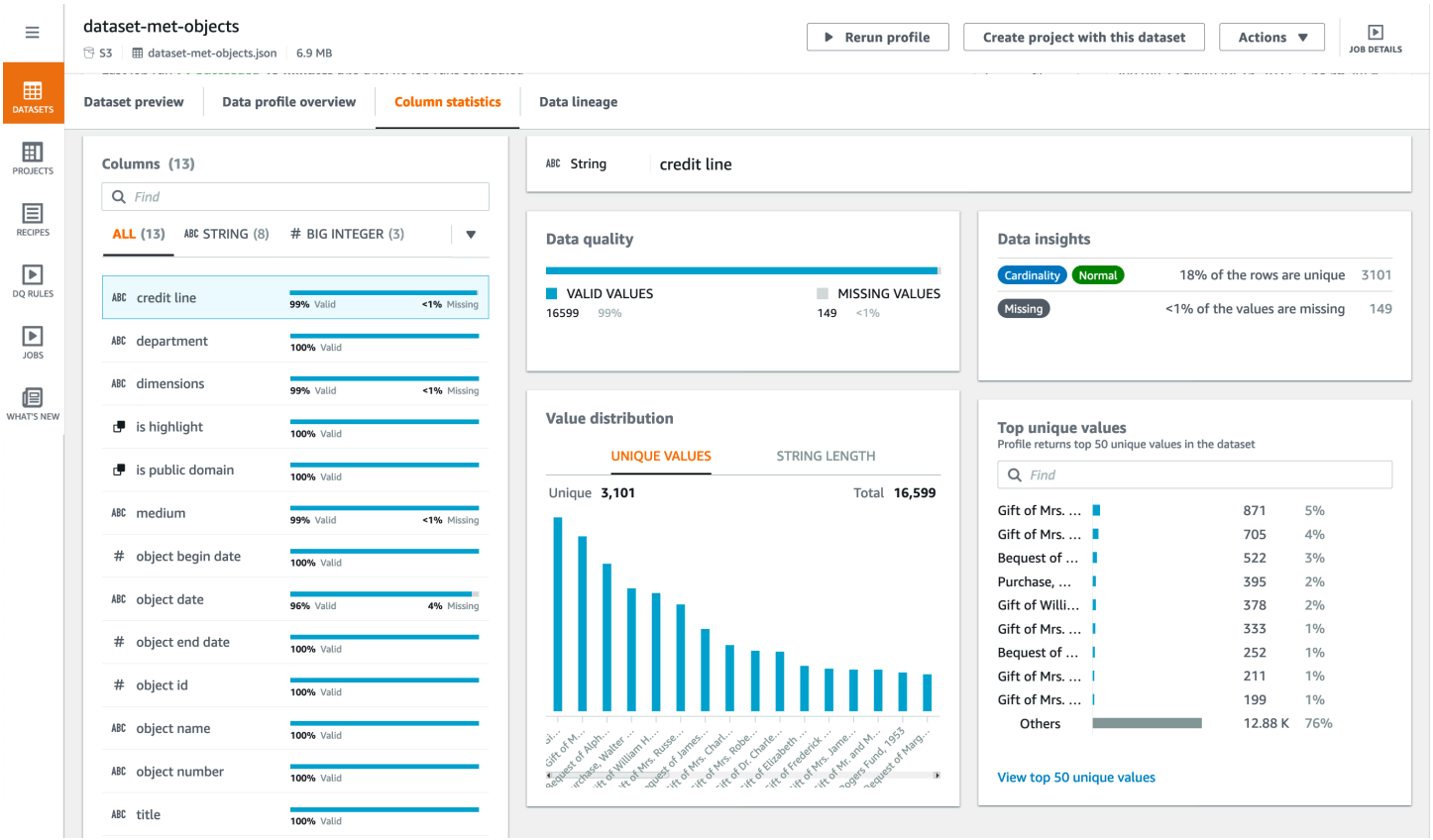


資料設定檔概觀 – 在此索引標籤上，您可以找到資料集統計資料和容積的圖形資料設定檔，如下所示。

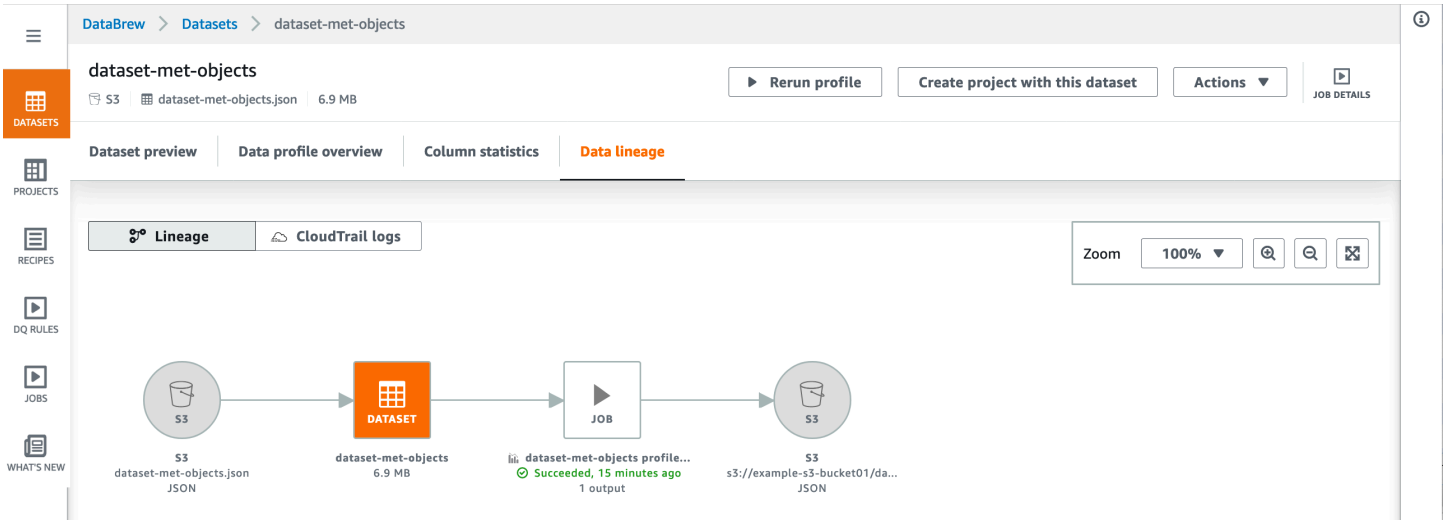


> **Note**
>
> 若要建立資料設定檔，請在資料集上執行 DataBrew 設定檔任務。如需如何進行該服務的詳細資訊，請參閱步驟 5：建立資料設定檔。

資料欄統計資料 – 在此索引標籤上，您可以找到資料集中每個資料欄的詳細統計資料，如下所示。



資料歷程 – 此標籤顯示資料集的建立方式及其在 DataBrew 中的使用方式的圖形表示，如下所示。



主題

- [刪除資料集]

刪除資料集

如果您不再需要資料集，可以將其刪除。刪除資料集不會對基礎資料來源造成任何影響。它只會移除 DataBrew 用來存取資料來源的資訊。

如果任何其他 DataBrew 資源依賴資料集，則無法刪除該資料集。例如，如果您目前有一個使用資料集的 DataBrew 專案，請先刪除專案，再刪除資料集。

若要刪除資料集，請從導覽窗格中選擇資料集。選擇您要刪除的資料集，然後針對動作選擇刪除。

連線至您的資料

如需連線至下列資料來源的詳細資訊，請選擇適用於您的區段。

- AWS Glue Data Catalog – 您可以使用 Data Catalog 來定義對存放在AWS雲端的資料物件的參考，包括下列服務：
  - Amazon Redshift
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS for MySQL
  - Amazon RDS for PostgreSQL

  DataBrew 會辨識已套用至 Data Catalog 資源的所有 Lake Formation 許可，因此 DataBrew 使用者只能在獲得授權的情況下存取這些資源。

  若要建立資料集，您可以指定 Data Catalog 資料庫名稱和資料表名稱。DataBrew 負責處理其他連線詳細資訊。
- AWS資料交換 – 您可以從AWS Data Exchange 中提供的數百個第三方資料來源中進行選擇。透過訂閱這些資料來源，您始終擁有up-to-date資料版本。

  若要建立資料集，您可以指定您訂閱或有權使用的資料交換資料產品的名稱。
- JDBC 驅動程式連線 – 您可以將 DataBrew 連線至 JDBC 相容資料來源來建立資料集。DataBrew 支援透過 JDBC 連線至下列來源：
  - Amazon Redshift
  - Microsoft SQL Server
  - MySQL

- Oracle
- PostgreSQL
- Snowflake

主題

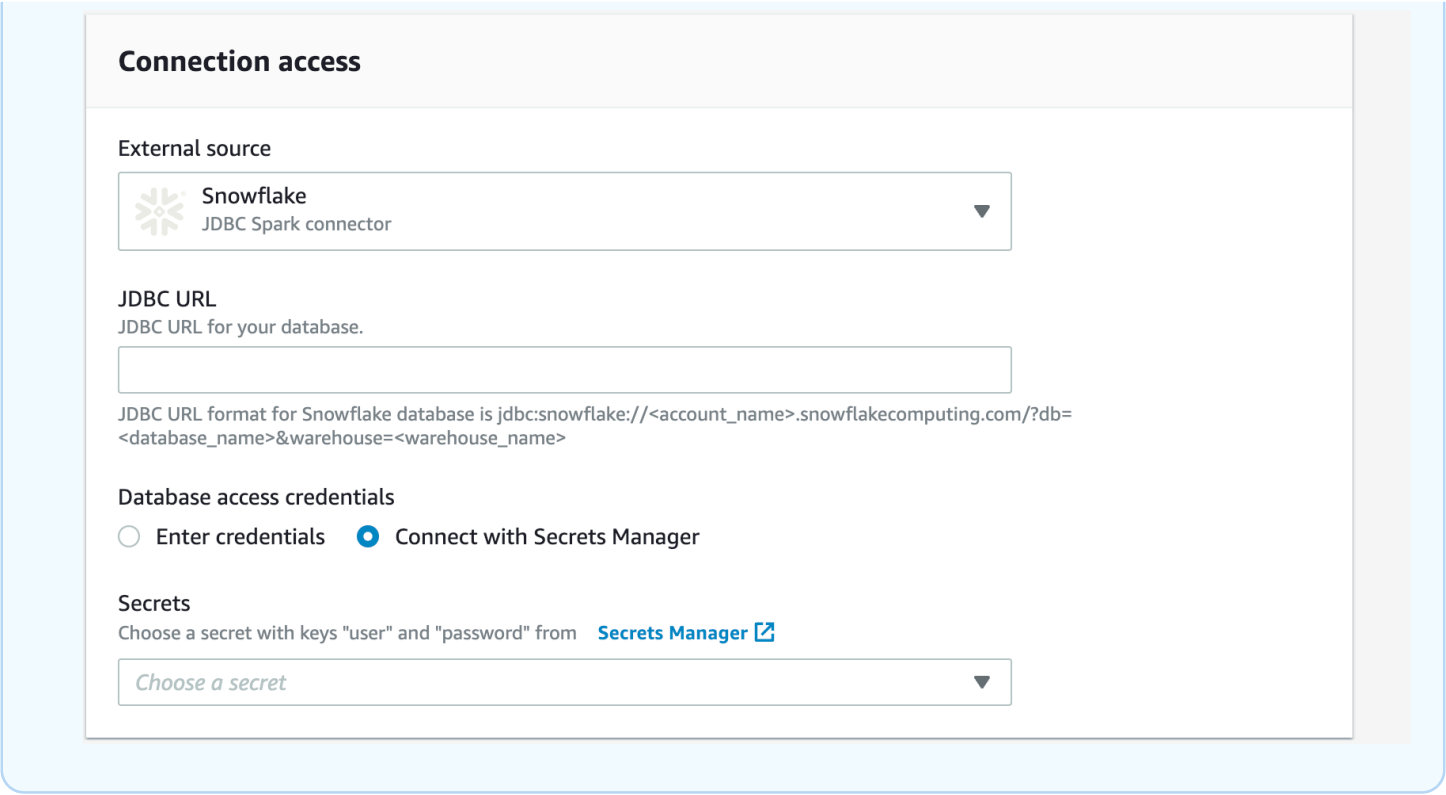
搭配 使用驅動程式AWS Glue DataBrew

資料庫驅動程式是實作資料庫連線通訊協定的檔案或 URL，例如 Java Database Connectivity (JDBC)。驅動程式可做為特定資料庫管理系統 (DBMS) 與另一個系統之間的轉接器或轉換器。

在這種情況下，它允許AWS Glue DataBrew連接到您的資料。然後，您可以從支援的資料來源存取資料庫物件，例如資料表或檢視。您使用的資料來源可能稱為資料庫、資料倉儲或其他項目。不過，基於本文件的目的，我們會將所有資料提供者稱為資料來源或連線。

若要使用 JDBC 驅動程式或 jar 檔案，請下載您需要的檔案，並將其放入 S3 儲存貯體。您用來存取資料的 IAM 角色需要具有兩個驅動程式檔案的讀取許可。

> **Note**
>
> With AWS Glue 4.0，原生支援連線至 Snowflake 做為資料來源。您不需要提供自訂`jar`檔案。在 中AWS Glue DataBrew，選擇 Snowflake 做為外部來源連線，並提供 Snowflake 執行個體的 URL。URL 將在表單 `https://account_identifier.snowflakecomputing.com` 中使用 hostname。
> 提供資料存取登入資料、Snowflake 資料庫名稱和 Snowflake 結構描述名稱。此外，如果您的 Snowflake 使用者沒有預設倉儲集，您將需要提供倉儲名稱。
> Snowflake 連線使用AWS Secrets Manager秘密來提供登入資料資訊。您在 中的專案和任務角色必須具有讀取此秘密的許可。



搭配 DataBrew 使用驅動程式

1. 使用 產品提供的 方法，了解您正在使用的資料來源版本。
2. 尋找所需的最新版本連接器和驅動程式。您可以在資料提供者網站上找到此資訊。
3. 下載 JDBC 檔案的必要版本。這些通常會儲存為 Java ARchives (.JAR) 檔案。
4. 您可以從主控台將驅動程式上傳至 S3 儲存貯體，或提供 .JAR 檔案的 S3 路徑。
5. 輸入基本連線詳細資訊，例如類別、執行個體等。
6. 輸入資料來源所需的任何其他組態資訊，例如虛擬私有雲端 (VPC) 資訊。

支援的 JDBC 驅動程式

產品	支援的版本	驅動程式指示和下載	支援的 SQL 查詢
	v6.x 或更新版本	Microsoft JDBC Driver for SQL Server	不支援

產品	支援的版本	驅動程式指示和下載	支援的 SQL 查詢
Microsoft SQL Server			
MySQL	v5.1 或更高版本	MySQL 連接器	不支援
Oracle	v11.2 或更新版本	Oracle JDBC 下載	不支援
PostgreSQL	v4.2.x 或更新版本	PostgreSQL JDBC 驅動程式	不支援
Amazon Redshift	v4.1 或更新版本	使用 JDBC 連線至 Amazon Redshift	支援
Snowflake	若要查看您的 Snowflake 版本，請使用 CURRENT ERSION，如 Snowflake 文件所述。	若要連線至 Snowflake，您需要下列兩項： • Snowflake JDBC 驅動程式 • 適用於 Spark 的 Snowflake 連接器	支援

若要連線到需要與 DataBrew 原生支援不同版本驅動程式的資料庫或資料倉儲，您可以提供您選擇的 JDBC 驅動程式。驅動程式必須與 JDK 8 或 Java 8 相容。如需如何尋找資料庫最新驅動程式版本的說明，請參閱 搭配 使用驅動程式AWS Glue DataBrew。

使用 DataBrew 連線至文字檔案中的資料

您可以為 DataBrew 支援的輸入檔案設定下列格式選項：

- 逗號分隔值 (CSV) 檔案
  - 分隔符號

    預設分隔符號是 .csv 檔案的逗號。如果您的檔案使用不同的分隔符號，請在建立資料集時，於其他組態區段中選擇 CSV 分隔符號的分隔符號。.csv 檔案支援下列分隔符號：
    - 逗號 (,)
    - Colon (:)
    - 分號 (;)
    - 管道 (|)
    - Tab (\t)
    - Caret (^)
    - 反斜線 (\)
    - 空格
  - 資料欄標頭值

    您的 CSV 檔案可以包含標頭列作為檔案的第一列。如果沒有，DataBrew 會為您建立標頭列。
    - 如果您的 CSV 檔案包含標頭列，請選擇將第一列視為標頭。如果您這樣做，CSV 檔案的第一列會被視為包含資料欄標頭值。
    - 如果您的 CSV 檔案不包含標頭列，請選擇新增預設標頭。如果您這麼做，DataBrew 會為 檔案建立標頭列，而且不會將第一列資料視為包含標頭值。DataBrew 建立的標頭包含 檔案中每一欄的底線和數字，格式為 `Column_1`、`Column_2`、 `Column_3`等。
- JSON 檔案

  DataBrew 支援兩種格式的 JSON 檔案：JSON Lines 和 JSON 文件。JSON Lines 檔案每行包含一個資料列。在 JSON 文件檔案中，所有資料列都包含在單一 JSON 結構或陣列中。您可以在建立 JSON 資料集時，在其他組態區段中指定您的 JSON 檔案類型。預設格式為 JSON Lines。
- Excel 檔案

下列適用於 DataBrew 中的 Excel 工作表：

- Excel 工作表載入

  根據預設，DataBrew 會載入 Excel 檔案中的第一個工作表。不過，您可以在建立 Excel 資料集時，在其他組態區段中指定不同的工作表號碼或工作表名稱。

- 資料欄標頭值

  您的 Excel 工作表可以包含標頭列作為檔案的第一列，但如果沒有，DataBrew 會為您建立標頭列。

  - 如果您的 Excel 工作表包含標頭列，請選擇將第一列視為標頭。如果您這樣做，則 Excel 工作表的第一列會被視為包含資料欄標頭值。
  - 如果您的 Excel 檔案不包含標頭列，請選擇新增預設標頭。透過這樣做，您可以指定 DataBrew 應為 檔案建立標頭列，而不是將第一列資料視為包含標頭值。DataBrew 建立的標頭包含 檔案中每一欄的底線和數字，格式為 `Column_1`、`Column_2`、 `Column_3`等。

在 Amazon S3 中連接多個檔案中的資料

使用 DataBrew 主控台，您可以導覽 Amazon S3 儲存貯體和資料夾，並為資料集選擇一個檔案。不過，資料集不需要限制為一個檔案。

假設您有一個名為 的 S3 儲存貯體`my-databrew-bucket`，其中包含名為 的資料夾`databrew-input`。在該資料夾中，假設您有許多 JSON 檔案，所有檔案都具有相同的檔案格式和副`.json`檔名。在 主控台上，您可以指定 的來源 URL`s3://my-databrew-bucket/databrew-input/`。在 DataBrew 主控台上，您可以選擇此資料夾。您的資料集包含該資料夾中的所有 JSON 檔案。

DataBrew 可以處理 S3 資料夾中的所有檔案，但前提是符合下列條件：

- 資料夾中的所有檔案都具有相同的格式。
- 資料夾中的所有檔案都有相同的副檔名。

如需支援的檔案格式和副檔名的詳細資訊，請參閱 DataBrew input formats。

使用多個檔案做為資料集時的結構描述

使用多個檔案做為 DataBrew 資料集時，所有檔案的結構描述都必須相同。否則，專案工作區會自動嘗試從多個檔案選擇其中一個結構描述，並嘗試將其他資料集檔案符合該結構描述。此行為會導致在專案工作區期間顯示的檢視不規律，因此任務輸出也會不規律。

如果您的檔案必須具有不同的結構描述，您需要建立多個資料集並分別進行描述。

使用 Amazon S3 的參數化路徑

在某些情況下，您可能想要使用遵循特定命名慣例的檔案建立資料集，或建立可以跨越多個 Amazon S3 資料夾的資料集。或者，您可能想要針對在 S3 位置中定期產生的相同結構化資料重複使用相同的資料集，路徑取決於特定參數。範例是名為 的路徑，用於資料生產日期。

DataBrew 支援此方法搭配參數化 S3 路徑。參數化路徑是包含規則表達式或自訂路徑參數或兩者的 Amazon S3 URL。

使用規則表達式以 S3 路徑定義資料集

路徑中的規則表達式對於比對一或多個資料夾的多個檔案很有用，同時篩選掉這些資料夾中不相關的檔案。

以下是幾個範例：

- 定義資料集，其中包含名稱開頭為 之資料夾的所有 JSON 檔案`invoice`。
- 定義包含資料夾中所有檔案的資料集，其名稱`2020`為 。

您可以在資料集 S3 路徑中使用規則表達式來實作此類方法。這些規則表達式可以取代 S3 URL 金鑰中的任何子字串 （但不能取代儲存貯體名稱）。

作為 S3 URL 中金鑰的範例，請參閱以下內容。這裡`my-bucket`是儲存貯體名稱，美國東部 （俄亥俄） 是AWS區域，而 `puppy.png`是金鑰名稱。

```
https://my-bucket.s3.us-west-2.amazonaws.com/puppy.png
```

在參數化 S3 路徑中，兩個角括號 (< 和 >) 之間的任何字元都會視為規則表達式。下列為兩個範例：

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json` 會在`databrew-input`名稱開頭為 的所有 子資料夾中`data.json`，比對名為 的所有檔案`invoice`。
- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/` 會比對資料夾中的所有檔案，其名稱`2020`為 。

在這些範例中， `.*`符合零個或多個字元。

> **Note**
>
> 您只能在 S3 路徑的索引鍵部分中使用規則表達式，也就是在儲存貯體名稱後面的部分。因此， `s3://my-databrew-bucket/<.*>-input/` 是有效的，但`s3://my-<.*>-bucket/<.*>-input/`不是。

我們建議您測試規則表達式，以確保它們只符合您想要的 S3 URL，而不是您不想要的 S3 URLs。

以下是規則表達式的一些其他範例：

- `<\d{2}>` 會比對由正好兩個連續數字組成的字串，例如 `07`或 `03`，但不是 `1a2`。
- `<[a-z]+.*>` 符合以一個或多個小寫拉丁字母開頭的字串，後面有零個或多個其他字元。範例為 `a3`、 `abc/def`或 `a-z`，但不是 `A2`。
- `<[^/]+>` 符合字串，其中包含除了斜線 (/) 以外的任何字元。在 S3 URL 中，斜線用於分隔路徑中的資料夾。
- `<.*=.*>` 符合包含等號 (=) 的字串`abc/day=2`，例如 `month=02`、 或 `=10`，但不符合 `test`。
- `<\d.*\d>` 符合以數字開頭和結尾的字串，且數字之間可以有任何其他字元，例如 `1abc2`、`01-02-03`或 `2020/Jul/21`，但不能有 `123a`。

使用自訂參數以 S3 路徑定義資料集

當您可能想要提供 S3 位置的參數時，使用自訂參數定義參數化資料集可提供優於使用規則表達式的優勢：

- 您可以使用規則表達式達到與 相同的結果，而不需要知道規則表達式的語法。您可以使用熟悉的術語定義參數，例如「開頭」和「包含」。
- 當您使用路徑中的參數定義動態資料集時，您可以在定義中包含時間範圍，例如「過去一個月」或「過去 24 小時」。如此一來，您的資料集定義稍後將與新的傳入資料搭配使用。

以下是一些您可能想要使用動態資料集的範例：

- 將依上次更新日期或其他有意義的屬性分割的多個檔案連接到單一資料集。然後，您可以將這些分割區屬性擷取為資料集中的額外資料欄。

- 將資料集中的檔案限制為符合特定條件的 S3 位置。例如，假設您的 S3 路徑包含日期型資料夾，例如 `folder/2021/04/01/`。在這種情況下，您可以參數化日期，並將其限制為特定範圍，例如「2021 年 3 月 1 日至 2021 年 4 月 1 日」或「上週」。

若要使用參數定義路徑，請定義參數，並使用下列格式將其新增至路徑：

```
s3://my-databrew-bucket/some-folder/{parameter1}/file-{parameter2}.json
```

> **Note**
>
> 如同 S3 路徑中的規則表達式，您只能在路徑的索引鍵部分使用參數，也就是在儲存貯體名稱後面的部分。

參數定義、名稱和類型中需要兩個欄位。類型可以是字串、數字或日期。日期類型的參數必須具有日期格式的定義，以便 DataBrew 可以正確解譯和比較日期值。或者，您可以定義參數的相符條件。您也可以選擇在 DataBrew 任務或互動式工作階段載入參數時，將參數的相符值作為資料欄新增至資料集。

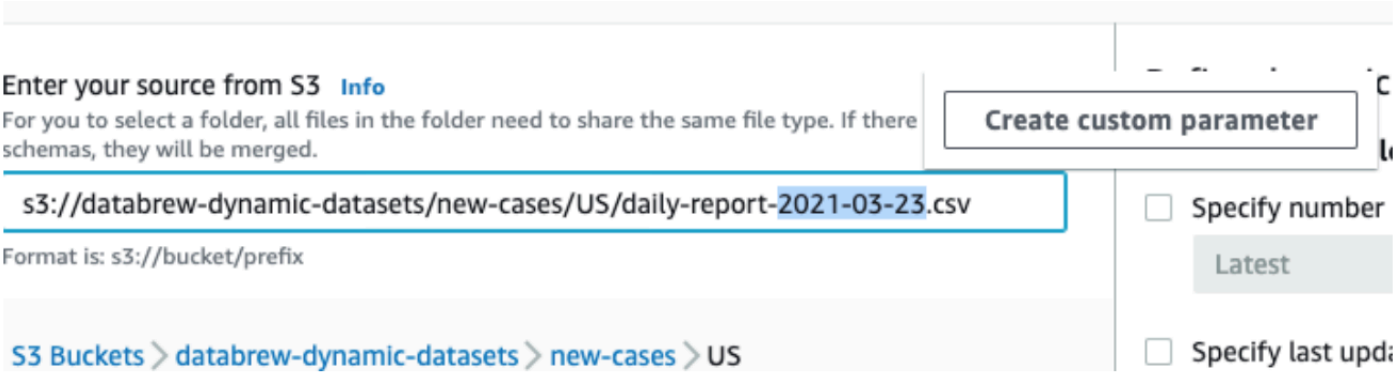
範例

讓我們考慮使用 DataBrew 主控台中的參數定義動態資料集的範例。在此範例中，假設輸入資料使用下列位置定期寫入 S3 儲存貯體：

- `s3://databrew-dynamic-datasets/new-cases/UR/daily-report-2021-03-30.csv`
- `s3://databrew-dynamic-datasets/new-cases/UR/daily-report-2021-03-31.csv`
- `s3://databrew-dynamic-datasets/new-cases/US/daily-report-2021-03-30.csv`
- `s3://databrew-dynamic-datasets/new-cases/US/daily-report-2021-03-31.csv`

這裡有兩個動態部分：國家/地區代碼，例如美國，以及檔案名稱中的日期，例如 2021-03-30。在這裡，您可以為所有檔案套用相同的清除配方。假設您想要每天執行清除任務。以下是您可以為此案例定義參數化路徑的方式：

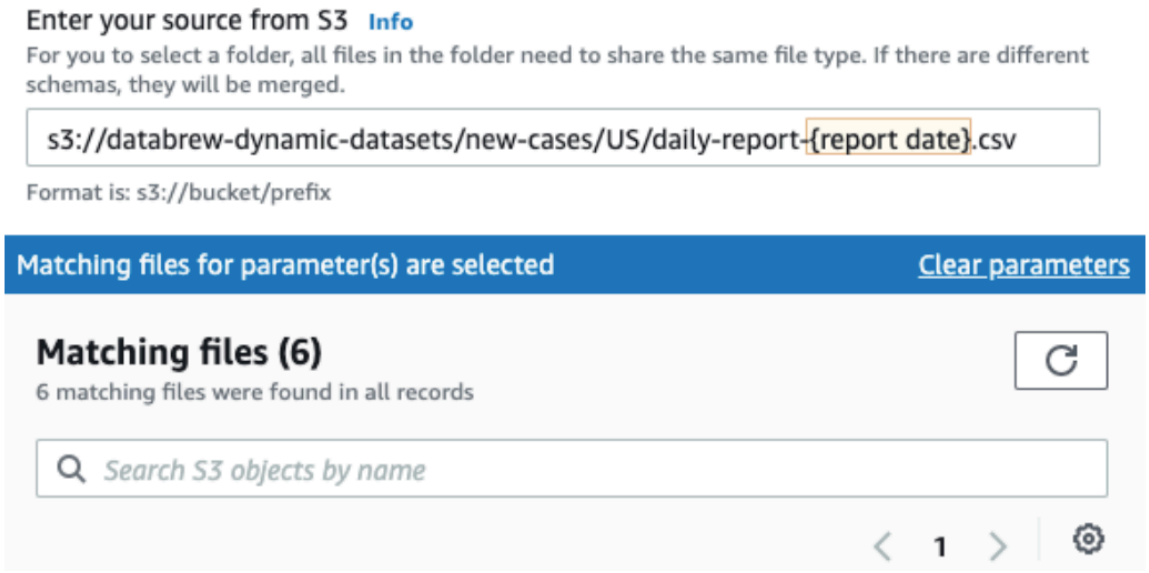
1. 導覽至特定檔案。
2. 然後選取不同的部分，例如日期，並以參數取代。在此情況下，請取代日期。



3. 開啟建立自訂參數的內容（按一下滑鼠右鍵）選單，並為其設定屬性：
   - 名稱：報告日期
   - 類型：日期
   - 日期格式：yyyy-MM-dd（從預先定義的格式中選取）
   - 條件（時間範圍）：過去 24 小時
   - 新增 為資料欄： true（已勾選）

   將其他欄位保留在其預設值。
4. 選擇建立。

完成後，您會看到更新的路徑，如下列螢幕擷取畫面所示。

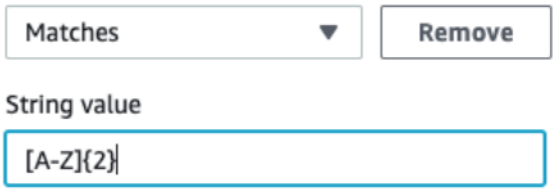


現在，您可以對國家/地區代碼執行相同的操作，並將其參數化，如下所示：

- 名稱：國家/地區代碼

- 類型：字串
- 新增 為資料欄： true （已勾選）

如果所有值都相關，則不需要指定條件。例如，在 new-cases 資料夾中，我們只有具有國家/地區代碼的子資料夾，因此不需要條件。如果您有其他要排除的資料夾，您可以使用下列條件。



此方法限制新案例的子資料夾包含兩個大寫拉丁字元。

在此參數化之後，您只有相符的資料集檔案，可以選擇建立資料集。

> **Note**
>
> 當您在條件中使用相對時間範圍時，載入資料集時會評估時間範圍。無論它們是預先定義的時間範圍，例如「過去 24 小時」或自訂時間範圍，例如「5 天前」，都是如此。此評估方法適用於互動式工作階段初始化期間或任務啟動期間載入的資料集。

選擇建立資料集後，您的動態資料集即可使用。例如，您可以先使用它來建立專案，並使用互動式 DataBrew 工作階段定義清除配方。然後，您可以建立排程每天執行的任務。此任務可能會在任務開始時，將清除配方套用至符合參數條件的資料集檔案。

動態資料集的支援條件

您可以使用參數或上次修改的日期屬性來篩選相符的 S3 檔案。

您可以在下面找到每個參數類型的支援條件清單。

與字串參數搭配使用的條件

DataBrew SDK 中的名稱	SDK 同義詞	DataBrew 主控台中的名稱	說明
是	常數，==	完全是	參數的值與條件中提供的值相同。

DataBrew SDK 中的名稱	SDK 同義詞	DataBrew 主控台中的名稱	說明
不是	not microSDHC， ！=	Is not	參數的值與條件中提供的值不同。
contains		Contains	參數的字串值包含條件中提供的值。
不包含		不包含	參數的字串值不包含條件中提供的值。
start_with		開頭為	參數的字串值以條件中提供的值開頭。
not start_with		開頭不是	參數的字串值開頭不是條件中提供的值。
end_with		Ends with	參數的字串值以條件中提供的值結尾。
not end_with		結尾不是	參數的字串值結尾不是條件中提供的值。
相符項目		相符	參數的值符合條件中提供的規則表達式。
不相符		不符合	參數的值不符合條件中提供的規則表達式。

Note

字串參數的所有條件都使用區分大小寫的比較。如果您不確定 S3 路徑中使用的案例，您可以使用「相符」條件搭配開頭為 的規則表達式值`(?i)`。這樣做會導致不區分大小寫的比較。例如，假設您希望字串參數以 開頭`abc`，但`AbcABC`也可以。在此情況下，您可以使用「相符」條件搭配 `(?i)^abc`做為條件值。

與數字參數搭配使用的條件

DataBrew SDK 中的名稱	SDK 同義詞	DataBrew 主控台中的名稱	說明
是	常數，==	完全是	參數的值與條件中提供的值相同。
不是	not microSDHC，！=	Is not	參數的值與條件中提供的值不同。
less_than	lt、<	Less than	參數的數值小於條件中提供的值。
less_than_equal	lte、<=	小於或等於	參數的數值小於或等於條件中提供的值。
greater_than	gt、>	Greater than	參數的數值大於條件中提供的值。
greater_than_equal	gte，>=	大於或等於	參數的數值大於或等於條件中提供的值。

與日期參數搭配使用的條件

DataBrew SDK 中的名稱	DataBrew 主控台中的名稱	條件值格式 (SDK)	說明
after	Start	ISO 8601 日期格式，例如 `2021-03-30T01:00:00Z` 或 `2021-03-30T01:00-07:00`	日期參數的值在條件中提供的日期之後。
before	結束	ISO 8601 日期格式，例如 `2021-03-30T01:00:00Z` 或 `2021-03-30T01:00-07:00`	日期參數的值在條件中提供的日期之前。

DataBrew SDK 中的名稱	DataBrew 主控台中的名稱	條件值格式 (SDK)	說明
relative_after	開始 （相對）	時間單位的正數或負數，例如 -48h或 +7d。	日期參數的值在條件中提供的相對日期之後。 載入資料集時，無論是初始化互動式工作階段或啟動相關聯的任務，都會評估相對日期。這是範例中稱為「現在」的時刻。
relative_before	結束 （相對）	時間單位的正數或負數，例如 -48h或 +7d。	日期參數的值在條件中提供的相對日期之前。 載入資料集時，無論是初始化互動式工作階段或啟動相關聯的任務，都會評估相對日期。這是範例中稱為「現在」的時刻。

如果您使用 SDK，請以下列格式提供相對日期：`±{number_of_time_units}{time_unit}`。您可以使用這些時間單位：

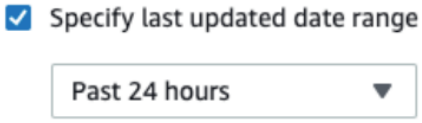
- -1h (1 小時前）
- +2d （從現在起 2 天）
- -120 公尺 (120 分鐘前）
- 5000 秒 （從現在起 5，000 秒）
- -3w (3 週前）
- +4M （從現在起 4 個月）
- -1y (1 年前）

載入資料集時，無論是初始化互動式工作階段或啟動相關聯的任務，都會評估相對日期。這是上述範例中稱為「現在」的時刻。

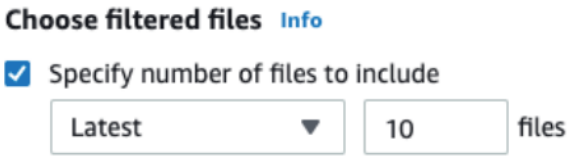
設定動態資料集的設定

除了提供參數化 S3 路徑之外，您還可以為具有多個檔案的資料集設定其他設定。這些設定會依其上次修改的日期篩選 S3 檔案，並限制檔案的數量。

與在路徑中設定日期參數類似，您可以在更新相符檔案時定義時間範圍，並僅將這些檔案包含在資料集中。您可以使用「2021 年 3 月 30 日」等絕對日期或「過去 24 小時」等相對範圍來定義這些範圍。



若要限制相符檔案的數量，請選取大於 0 的檔案數量，以及您想要最新或最舊的相符檔案。



資料類型

資料集每個資料欄的資料會轉換為下列其中一種資料類型：

- 位元組 – 1 位元組帶正負號的整數。數字的範圍是 -128 到 127。
- short – 2 位元組帶正負號整數。數字的範圍是 -32768 到 32767。
- integer – 4 位元組帶正負號整數。數字的範圍是從 -2147483648 到 2147483647。
- long – 8 位元組帶正負號整數。數字的範圍是 -9223372036854775808 到 9223372036854775807。
- float – 4 位元組的單精度浮點數。
- 雙 – 8 位元組雙精度浮點數。
- 十進位 – 帶正負號的小數，總計最多 38 位數，小數點後最多 18 位數。
- string – 字元字串值。
- 布林值 – 布林值類型有兩種可能的值：`true` 和 `false` 或 `yes` 和 `no`。
- 時間戳記 – 包含欄位年、月、日、小時、分鐘和秒的值。
- date – 包含欄位年份、月份和日期的值。

進階資料類型

進階資料類型是 DataBrew 在專案的字串欄中偵測到的資料類型，因此不屬於資料集。如需進階資料類型的資訊，請參閱進階資料類型。

進階資料類型

進階資料類型是 DataBrew 透過模式比對在專案的字串欄中偵測到的資料類型。當您按一下字串欄時，如果欄中 50% 或更多的值符合該資料類型的條件，則該欄會標記為對應的進階資料類型。

DataBrew 可以偵測的資料類型為：

- 日期/時間戳記
- SSN
- 電話號碼
- Email
- 信用卡
- Gender
- IP 位址
- URL
- 郵遞區號
- Country
- Currency
- State
- 城市

您可以使用下列轉換來使用進階資料類型：

- GET_ADVANCED_DATATYPE：假設有字串欄， 會識別該欄的進階資料類型。
- EXTRACT_ADVANCED_DATATYPE_DETAILS：擷取進階資料類型的詳細資訊。
- ADVANCED_DATATYPE_FILTER：根據進階資料類型偵測篩選目前的來源資料欄。
- ADVANCED_DATATYPE_FLAG：根據目前來源資料欄的值建立新的旗標資料欄。

在 中驗證資料品質AWS Glue DataBrew

為了確保資料集的品質，您可以在規則集中定義資料品質規則清單。規則集是一組規則，可將不同的資料指標與預期值進行比較。如果不符合任何規則的條件，則整體規則集會驗證失敗。然後，您可以檢查每個規則的個別結果。對於導致驗證失敗的任何規則，您可以進行必要的更正和重新驗證。

規則的範例包括下列項目：

- 資料欄中的值"APY"介於 0 到 100 之間
- 資料欄中遺失值的數量group_name不超過 5%

您可以為個別資料欄定義每個規則，或將其獨立套用至數個選取的資料欄，例如：

- 資料欄 "rate"、"pay"、 的最大值不超過 100"increase"。

規則可以包含多個簡單的檢查。您可以定義所有應該是 true 還是任何，例如：

- 資料欄中的值"ProductId"應以資料欄中值的 "asin-" AND 長度"ProductId"為 32 開頭。

您可以針對彙總值驗證規則max，例如 min、 或 number of duplicate values，其中只比較一個值，或是資料欄每一列中的非彙總值。在後一種情況下，您也可以定義「傳遞」閾值，例如 value in columnA > value in columnB for at least 95% of rows。

如同設定檔資訊，您只能為簡單類型的資料欄定義資料欄層級資料品質規則，例如字串和數字。您無法定義複雜類型資料欄的資料品質規則，例如陣列或結構。如需使用設定檔資訊的詳細資訊，請參閱 建立和使用AWS Glue DataBrew設定檔任務。

驗證資料品質規則

定義規則集之後，您可以將其新增至設定檔任務以進行驗證。您可以為資料集定義多個規則集。

例如，一個規則集可能包含最低可接受條件的規則。該規則集的驗證失敗可能表示資料無法接受進一步使用。範例是用於機器學習訓練之資料集的索引鍵資料欄中缺少值。您可以使用第二個規則集搭配更嚴格的規則，來驗證資料集的品質是否良好，因此不需要清除。

您可以在設定檔任務組態中套用為指定資料集定義的一或多個規則集。當設定檔任務執行時，除了資料設定檔之外，還會產生驗證報告。驗證報告可在與設定檔資料相同的位置取得。如同設定檔資訊，您可

以在 DataBrew 主控台中探索結果。在資料集詳細資訊檢視中，選擇資料品質索引標籤以檢視結果。如需使用設定檔資訊的詳細資訊，請參閱 建立和使用AWS Glue DataBrew設定檔任務。

對驗證結果採取行動

當 DataBrew 設定檔任務完成時，DataBrew 會傳送 Amazon CloudWatch 事件，其中包含該任務執行的詳細資訊。如果您也設定任務來驗證資料品質規則，DataBrew 會為每個已驗證的規則集傳送事件。事件包含其結果 (SUCCEEDED、 FAILED或 ERROR)，以及詳細資料品質驗證報告的連結。然後，您可以根據驗證狀態調用下一個動作來自動執行進一步的動作。如需將事件連線至目標動作的詳細資訊，例如 Amazon SNS 通知、AWS Lambda函數叫用等，請參閱 Amazon EventBridge 入門。

以下是 DataBrew 驗證結果事件的範例：

```
{
  "version": "0",
  "id": "fb27348b-112d-e7c2-560d-85e7c2c09964",
  "detail-type": "DataBrew Ruleset Validation Result",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2021-11-18T13:15:46Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "datasetName": "MyDataset",
    "jobName": "MyProfileJob",
    "jobRunId": "db_f07954d20d083de0c1fc1eee11498d8635ee5be4ca416af27d33933e91ff4e6e",
    "rulesetName": "MyRuleset",
    "validationState": "FAILED",
    "validationReportLocation": "s3://MyBucket/MyKey/
MyDataset_f07954d20d083de0c1fc1eee11498d8635ee5be4ca416af27d33933e91ff4e6e_dq-
validation-report.json"
  }
}
```

您可以使用 等事件的屬性detail-type，source以及 detail 屬性的巢狀屬性，在 Amazon Eventbridge 中建立事件模式。例如，符合任何 DataBrew 任務中所有失敗驗證的事件模式如下所示：

```
{
  "source": ["aws.databrew"],
  "detail-type": ["DataBrew Ruleset Validation Result"],
```

```
  "detail": {
   "validationState": ["FAILED"]
  }
}
```

如需建立規則集和驗證其規則的範例，請參閱 使用資料品質規則建立規則集。如需在 DataBrew 中使用 CloudWatch 事件的詳細資訊，請參閱 使用 CloudWatch Events 自動化 DataBrew

使用資料品質規則建立規則集

在下列程序中，您可以找到建立規則集並將其套用至資料集的範例。規則集是一組規則，可將不同的資料指標與預期值進行比較。然後，您可以在設定檔任務中使用此規則集來驗證其中包含的資料品質規則。

使用資料品質規則建立範例規則集

1. 登入AWS 管理主控台並開啟 DataBrew 主控台，網址為 https：//https://console.aws.amazon.com/databrew/。
2. 從導覽窗格中選擇 DQ 規則，然後選擇建立資料品質規則集。
3. 輸入規則集的名稱。或者，輸入規則集的描述。
4. 在關聯的資料集下，選擇要與規則集建立關聯的資料集。

   選取資料集之後，您可以檢視右側的資料集預覽窗格。
5. 當您決定要建立的資料品質規則時，請使用資料集預覽窗格中的預覽來探索資料集的值和結構描述。預覽可讓您深入了解資料可能遇到的潛在問題。

   有些資料來源，例如資料庫，不支援資料預覽。在這種情況下，您可以執行設定檔任務，而無需先驗證資料品質規則。然後，您可以使用資料描述檔取得資料結構描述和值分佈的相關資訊。
6. 檢查建議索引標籤，其中列出您可以在建立規則集時使用的一些規則建議。您可以選取所有、部分或無任何建議。

   選取相關建議後，選擇新增至規則集。

   這會將規則新增至您的規則集。視需要檢查和修改參數。請注意，資料品質規則中只能使用字串、數字和布林值等簡單類型的資料欄。
7. 選擇新增另一個規則以新增建議未涵蓋的規則。您可以變更規則名稱，以便稍後更輕鬆地解譯驗證結果。

8. 使用資料品質檢查範圍來選擇是否要針對此規則中的每個檢查選取個別資料欄，或是否應套用到您選取的資料欄群組。例如，如果您的資料集有數個數值資料欄的值應該介於 0 到 100 之間，您可以定義規則一次，然後選取所有要由此規則檢查的資料欄。
9. 如果您的規則會有多個檢查，請在規則成功條件下拉式清單中，選擇是否應符合所有檢查，還是哪些檢查符合條件。
10. 選取將在資料品質檢查下拉式清單中執行以驗證此規則的檢查。如需可用檢查的詳細資訊，請參閱可用的檢查。
11. 如果您為資料品質檢查範圍中的每個資料欄選擇個別檢查，請選擇資料欄。 選取或輸入此檢查的資料欄名稱。
12. 根據檢查選取參數。有些條件只接受提供的自訂值，有些條件也支援參考另一個資料欄。
13. 如果您選擇檢查資料欄值，例如包含字串值的條件，則可以指定「傳遞」閾值。例如，如果您希望至少 95% 的值滿足條件，則需要選擇大於等於閾值的條件，輸入 95 做為閾值，並在閾值區段的下一個下拉式清單中保留「增加百分比） 資料列」。或者，如果您想要不超過 10 個資料列，其中值缺少條件為 true，則可以選取小於等於條件，在閾值中輸入 10，然後在下一個下拉式清單中選擇資料列。請注意，如果您在驗證期間使用不同大小的範例，可能會得到不同的結果。
14. 視需要新增更多規則。
15. 選擇建立規則集。

使用規則集建立設定檔任務

如前所述建立規則集後，系統會將您導向資料品質規則頁面，其中會顯示您帳戶中的所有規則集。

建立包含規則集的設定檔任務

1. 選擇您先前建立的規則集名稱，以檢視其詳細資訊。
2. 選擇使用規則集建立設定檔任務。

   任務名稱會自動填入，但您可以視需要進行變更。
3. 對於任務執行範例，您可以選擇執行整個資料集或有限數量的資料列。

   如果您選擇執行有限的樣本大小，請注意，對於某些規則，結果可能會與完整資料集不同。
4. 針對任務輸出設定，選擇任務輸出的 S3 位置。選擇您能夠存取之具名 Amazon S3 儲存貯體中的任何資料夾。如果您為此儲存貯體輸入不存在的資料夾名稱，則會建立此資料夾。

   成功完成設定檔任務後，此資料夾將包含 JSON 格式的資料和資料品質規則驗證報告的設定檔。
5. 在資料品質規則下，請注意，您的規則集會列在資料品質規則集名稱下。

6. 在許可下，選取或建立角色，以授予 DataBrew 從輸入 Amazon S3 位置讀取和寫入任務輸出位置的存取權。如果您沒有就緒的角色，請選取建立新 IAM 角色。
7. 視需要修改任何其他選用設定建立和使用AWS Glue DataBrew設定檔任務，如 中所述。
8. 選擇建立並執行任務。

檢查資料品質規則的驗證結果並更新資料品質規則

設定檔任務完成後，您可以檢視資料品質規則的驗證結果，並視需要更新規則。

檢視資料品質規則的驗證資料

1. 在 DataBrew 主控台上，選擇檢視資料設定檔。這樣做會顯示資料集的資料設定檔概觀索引標籤。
2. 選擇資料品質規則索引標籤。在此索引標籤上，您可以檢視所有資料品質規則的結果。
3. 選取個別規則以取得該規則的詳細資訊。

對於驗證失敗的任何規則，您可以進行必要的更正。

更新資料品質規則

1. 在導覽窗格中，選擇 DQ RulesS。
2. 在資料品質規則集名稱下，選擇包含您計劃編輯之規則的資料集。
3. 選擇您要變更的規則，然後選擇編輯。
4. 進行必要的更正，然後選擇更新規則集。
5. 重新執行任務。重複此程序，直到所有驗證通過為止。

可用的檢查

下表列出可用於規則的所有可用條件的參考。請注意，彙總條件不能與相同規則中的非彙總條件結合。

> **Note**
>
> 對於 SDK 使用者，若要將相同的規則套用至多個資料欄，請使用規則的 ColumnSelectors 屬性，並使用其名稱或規則表達式指定已驗證的資料欄。在此情況下，您應該使用隱含 CheckExpression。例如，“> :val”將每個所選欄中的值與提供的值進行比較。DataBrew 使用隱含語法來定義動態資料集中的 FilterExpression。如果您想要為每個檢查個別指定資

料欄 (ColumnSelectors)，請勿設定 ColumnSelectors 屬性。反之，請提供明確表達式。例如，“:col > :val”作為規則中的 CheckExpression。

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
彙總資料集條件	資料列數		與自訂值的數值比較	"CheckExpression": "AGG(ROWS_COUNT) > :val", "SubstitutionMap": {":val", "10000"}
	欄數		與自訂值的數值比較	"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}
	重複的資料列		與自訂值的數值比較	"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT) < :val", "SubstitutionMap": {":val", "100"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
				或 "CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
彙總資料欄統計資料條件	缺少值		與自訂值的數值比較	"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"} 或 "CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	重複值		與自訂值的數值比較	"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"} 或 "CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	有效值		與自訂值的數值比較	"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"} 或 "CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	不同的值		與自訂值的數值比較	"CheckExpression": "AGG(DISTINCT_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "1000"} 或 "CheckExpression": "AGG(DISTINCT_VALUES_PERCENTAGE) >= :val", "SubstitutionMap": {":val", "50"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	唯一值		與自訂值的數值比較	"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"} 或 "CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	極端值	Z 分數閾值	與自訂值的數值比較	"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"} 或 "CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例

	值分佈統計資料	統計資料名稱（請參閱下表）	與自訂值的數值比較	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				或
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				如需可能STAT_NAME 的值，請參閱下表

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例

	數值統計資料	統計資料名稱（請參閱下表）	與自訂值的數值比較	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				或
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				如需可能STAT_NAME 的值，請參閱下表

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
非彙總（接受閾值）	值完全為		與值清單的精確比較	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	值不完全		值不應完全符合清單中的任何值	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	字串值		與自訂值或其他字串欄的字串比較	"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"} 或 "CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	數值		與自訂值或其他數值欄的數值比較	"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"} 或 "CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}

條件類型	資料品質檢查	額外參數	比較類型	SDK 語法範例
	值字串長度		與自訂值或其他數值欄的數值比較	"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"} 或 "CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}

數值比較

DataBrew 支援下列數值比較操作：等於 (==)、不等於 (！=)、小於 (<)、小於等於 (<=)、大於 (>)、大於等於 (>=) 和介於 (is_介於 ：val1 和 ：val2 之間 ） 之間。

字串比較

支援下列字串比較：開頭、開頭、結尾、結尾、結尾、包含、不包含、等於、不等於、相符、不相符。

下表顯示可用於值分佈統計資料和數值統計資料的可用統計資料：

資料品質檢查	統計資料名稱	額外參數	SDK 語法
值分佈統計資料	最少		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	最多		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	中位數		`"CheckExpression": "AGG(MEDIAN) >= :val", "SubstitutionMap": {":val", "50"}`
	Mean		`"CheckExpression": "AGG(MEAN) <= :val",`

資料品質檢查	統計資料名稱	額外參數	SDK 語法
			`"Substitu tionMap": {":val", "10"}`
	Mode		`"CheckExp ression": "AGG(MODE ) > :val", "Substitu tionMap": {":val", "0"}`
	標準偏差		`"CheckExp ression": "AGG(STAN DARD_DEVI ATION) > :val", "Substitu tionMap": {":val", "0"}`
	Entropy		`"CheckExp ression": "AGG(ENTR OPY) > :val", "Substitu tionMap": {":val", "0"}`

資料品質檢查	統計資料名稱	額外參數	SDK 語法
數值統計資料	總和		"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}
	峰度		"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}
	偏斜		"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}
	變異數		"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}

資料品質檢查	統計資料名稱	額外參數	SDK 語法
	絕對偏差		"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}
	分位數	分位數：'0.25'、'0.5'、'0.75' 之一	"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}

建立和使用AWS Glue DataBrew專案

在 中AWS Glue DataBrew，專案是資料分析和轉換工作的中心部分。

建立專案時，您會將兩個基本元件結合在一起：

- 資料集，提供對來源資料的唯讀存取。如需詳細資訊，請參閱使用 連線至資料AWS Glue DataBrew。
- 將 DataBrew 資料轉換套用至資料集的配方。如需詳細資訊，請參閱建立和使用AWS Glue DataBrew配方。

DataBrew 主控台以高度互動、直覺式的使用者介面呈現您的專案。它鼓勵您實驗數百個資料轉換，因此您可以了解它們的運作方式，以及它們對您的資料有何影響。

您在專案檢視中看到的資料是資料集的範例。由於資料集可能非常大，有數千或甚至數百萬個資料列，使用範例有助於確保 DataBrew 主控台在您以各種方式轉換範例資料時保持回應。根據預設，範例包含來自資料集的前 500 列資料。您可以為樣本大小選擇不同的設定，以及選擇哪些資料列。

當您轉換範例資料時，DataBrew step-by-step 可協助您建置和精簡專案配方，這是您目前套用的逐步轉換系列。您的work-in-progress配方會自動儲存，因此您可以隨時離開專案檢視，稍後再返回並從您停止的地方挑選。

當您的配方可供使用時，您可以發佈它。發佈配方可讓 DataBrew 任務子系統使用，您可以在其中將配方套用至整個資料集，或建立廣泛的資料設定檔，讓您了解資料的結構、內容和統計特性。

主題

建立專案

使用下列程序來建立專案。

建立專案

1. 登入AWS 管理主控台並開啟 DataBrew 主控台 。

2. 在導覽窗格中，選擇專案。然後選擇建立專案。
3. 輸入專案的名稱。然後選擇要連接到專案的配方：
   - 如果您是從頭開始，請選擇建立新配方。這樣做會建立新的空白配方，並將其連接到您的專案。
   - 如果您有先前發佈的配方要用於此專案，請選擇編輯現有配方。如果配方目前連接到另一個專案，或已為其定義任何任務，則您無法在新專案中使用它。選擇瀏覽配方以查看可用的配方。
   - 如果您有先前已發佈且想要匯入其步驟的現有配方，請從配方中選擇匯入步驟，然後執行下列動作：
     1. 選擇瀏覽配方以查看可用的配方。
     2. 選擇您要使用的配方發佈版本。配方可以有多個版本，取決於您在專案檢視中發佈它的頻率。
     3. 選擇檢視配方步驟以檢查配方中的資料轉換。
4. 在您擁有配方之後，請在選取資料集窗格中選擇要使用的資料集：
   - 我的資料集 – 選擇您先前建立的資料集。如需詳細資訊，請參閱 建立專案)。
   - 範例檔案 – 根據 維護的範例資料建立新的資料集AWS。此範例資料是探索 DataBrew 可以做什麼的好方法，而無需提供您自己的資料。請務必輸入資料集的名稱。
   - 新資料集 – 建立新的資料集。如需詳細資訊，請參閱建立專案。
5. 針對存取許可，選擇允許 DataBrew 從 Amazon S3 輸入位置讀取的AWS Identity and Access Management(IAM) 角色。對於AWS您的帳戶擁有的 S3 位置，您可以選擇`AwsGlueDataBrewDataAccessRole`服務受管角色。這樣做可讓 DataBrew 存取您擁有的 S3 資源。
6. 在取樣窗格中，您可以找到 DataBrew 從資料集建置資料範例的選項。

   針對類型，選擇 DataBrew 應如何從您的資料集取得資料列：
   - 使用前 n 個資料列，根據資料集中的前 列建立範例。
   - 使用隨機資料列，根據資料集中的資料列隨機選取來建立範例。
   - 選擇要出現在範例中的資料列數目：500、1，000、2，500 或自訂樣本大小，最多 5，000 個資料列。較小的範例大小可讓 DataBrew 更快地執行轉換，節省您開發配方的時間。較大的樣本大小更準確地反映基礎來源資料的組成。不過，專案工作階段初始化和互動式轉換速度較慢。
7. （選用）選擇標籤以將標籤連接至資料集。

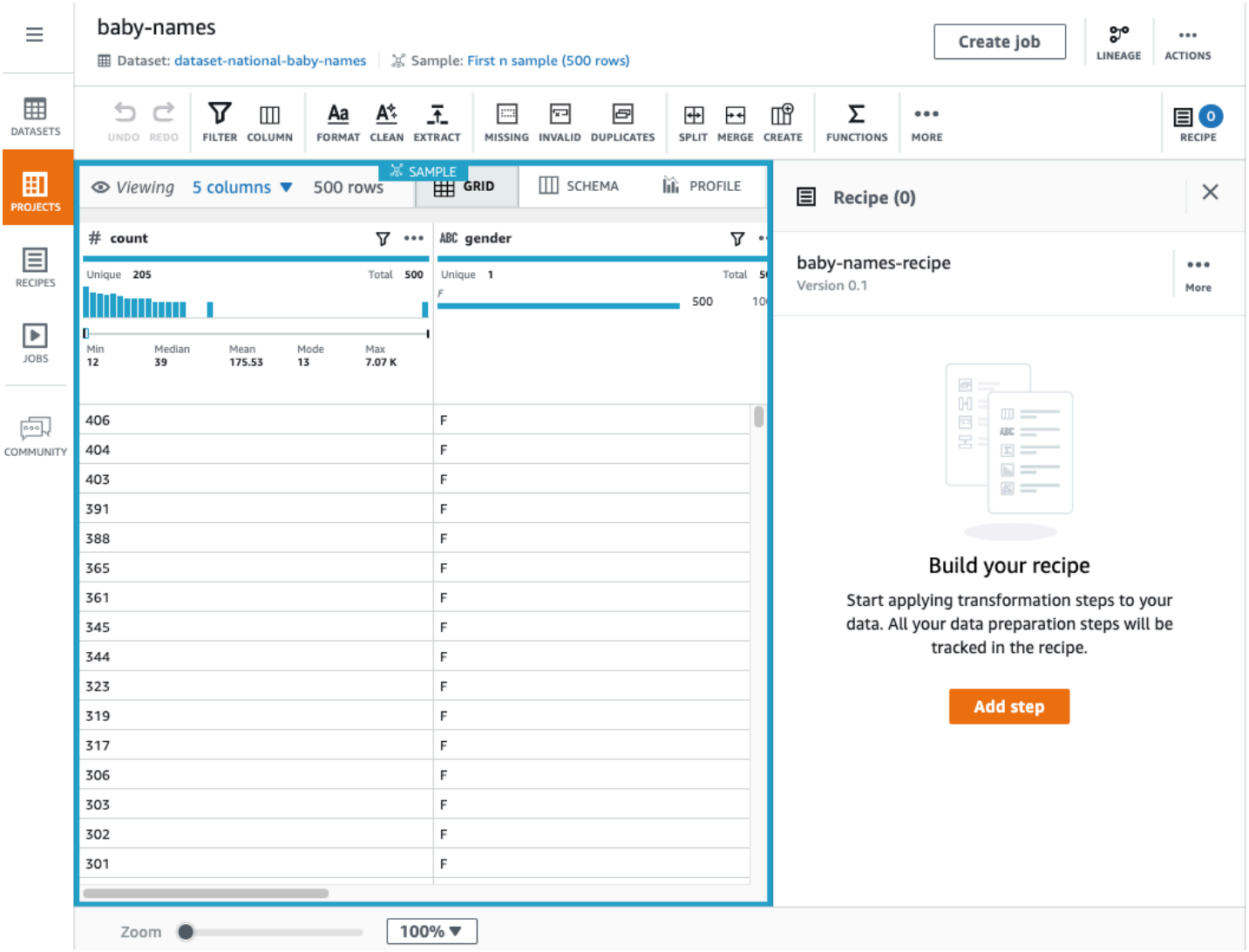
   標籤是簡單的標籤，由使用者定義的金鑰和選用值組成，可讓您根據用途、擁有者、環境或其他條件更輕鬆地管理、搜尋和篩選 DataBrew 專案。

8. 設定如您想要，請選擇建立任務。

DataBrew 會視需要建立新的資料集、視需要建立新的配方、建置資料範例，以及建立互動式專案工作階段。此程序可能需要幾分鐘的時間才能完成。當專案可供使用時，您可以開始使用資料範例。

DataBrew 專案工作階段概觀

在 DataBrew 專案工作階段中，您會在互動式工作區內工作。



左側窗格會顯示資料的目前檢視。右窗格顯示專案的轉換配方，目前為空白。

在資料網格的右上角，有三個索引標籤：GRID、 SCHEMA和 PROFILE。選擇其中一個索引標籤會在工作區中顯示對應的檢視；接下來會說明這些檢視。

網格檢視

網格檢視是預設檢視，其中範例會以表格格式顯示。使用下列程序進行網格檢視的簡短演練。

若要逐步解說網格檢視

1. 首先檢視整個空間：
   a. 左右捲動以查看所有欄。
   b. 上下捲動以查看所有資料值。
   c. 使用工作區底部的縮放控制項來調整網格的放大等級。
2. 在右上角，檢視顯示多少個範例資料欄，以及範例中目前的資料列數。

   若要變更顯示的資料欄，請選擇 N 資料欄連結 （其中 N 是目前顯示的資料欄數）。選擇您想要的資料欄，然後選擇顯示選取的資料欄。
3. 現在您可以開始實驗 DataBrew 轉換。請嘗試以下做法：
   a. 從轉換工具列中，選擇選擇格式、變更為大寫。
   b. 針對來源資料欄，選擇包含字元資料的資料欄。
   c. 將其他設定保留為各自的預設設定。
   d. 若要查看轉換後的資料會是什麼樣子，請選擇預覽變更。然後，若要將此轉換新增至您的配方，請選擇套用。

   每當您套用資料轉換時，DataBrew 都會將其新增至配方的工作副本。這會出現在工作區的右側。
4. 請嘗試以下做法：
   a. 從轉換工具列中，根據函數選擇建立。
   b. 針對選取函數，選擇 SQUARE ROOT。
   c. 針對來源資料欄，選擇包含數值資料的資料欄。
   d. 將其他設定保留為預設值。
   e. 選擇預覽變更，以查看轉換後的資料外觀。然後，若要將此轉換新增至您的配方，請選擇套用。
5. 選擇 RECIPE，摺疊右上角的配方窗格。若要展開配方窗格，請再次選擇 RECIPE。

發佈您配方的新版本

隨著您繼續套用轉換，配方中的步驟數量會增加。您可以隨時發佈新版本的配方。發佈配方可在 DataBrew 的其他位置使用。透過這樣做，您可以執行配方任務來轉換整個資料集，而不是只轉換專案資料範例。

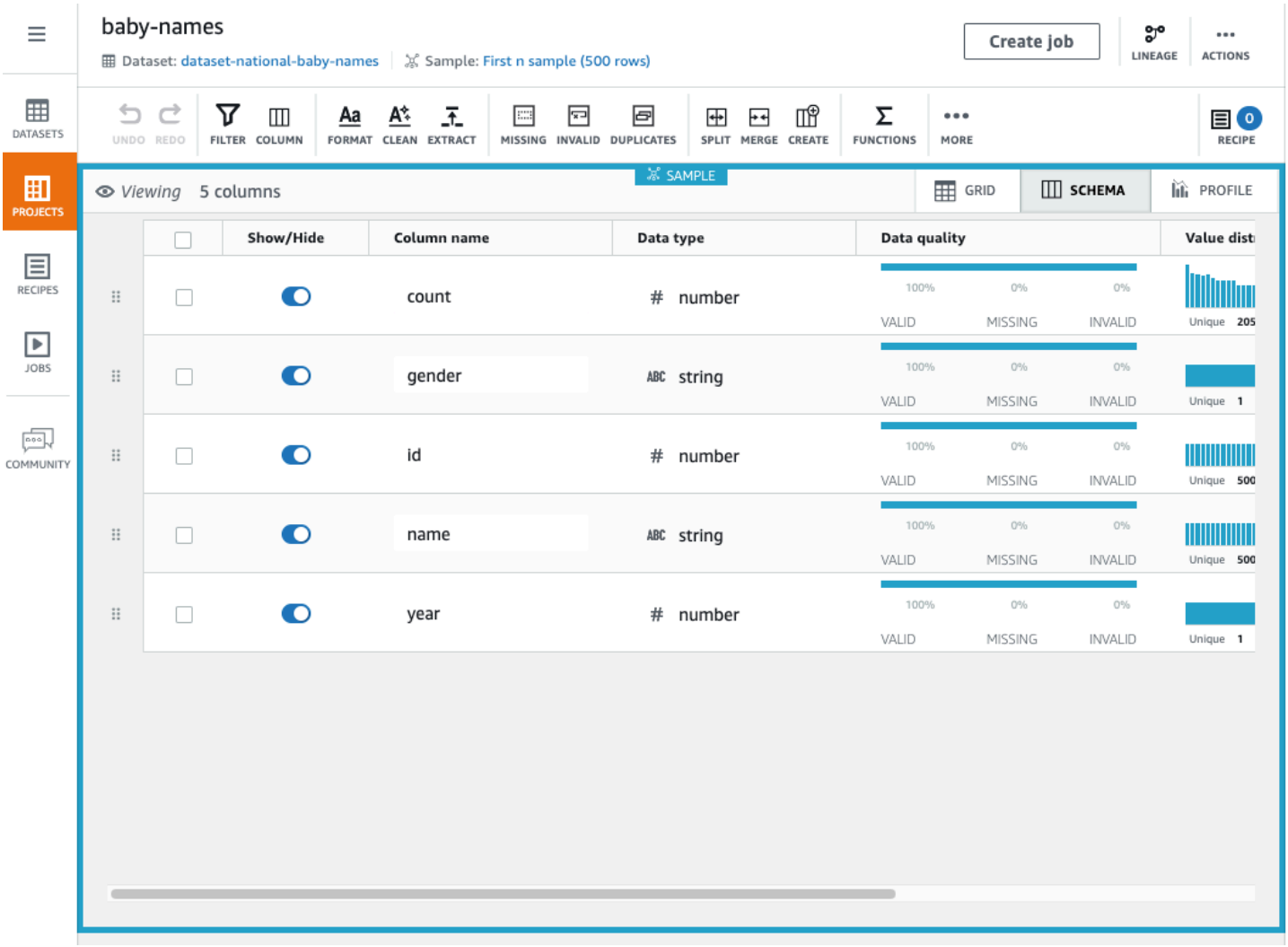
發佈配方也鼓勵配方開發的增量式反覆方法：您可以隨需發佈配方的新版本，以便在需要時返回「上次已知良好」配方版本。

發佈新版本的配方

- 在配方窗格中，選擇發佈。輸入此版本配方的描述，然後選擇發佈。

結構描述檢視

如果您選擇 SCHEMA 標籤，檢視會變更，如以下螢幕擷取畫面所示。



在結構描述檢視中，您可以查看每個資料欄中資料值的統計資料。

在最左欄的顯示/隱藏旁，選擇任何資料欄。資料欄詳細資訊窗格會出現在右側。此窗格顯示資料欄值的統計資料摘要。

您可以輸入資料欄名稱的新名稱來重新命名資料欄。

您可以透過拖放資料欄來重新排列資料欄順序。

設定檔檢視

如果您選擇 PROFILE 索引標籤，您可以看到有關專案的詳細容積資訊。執行此操作之前，您可以執行 DataBrew 任務來建立設定檔。

若要逐步解說設定檔檢視

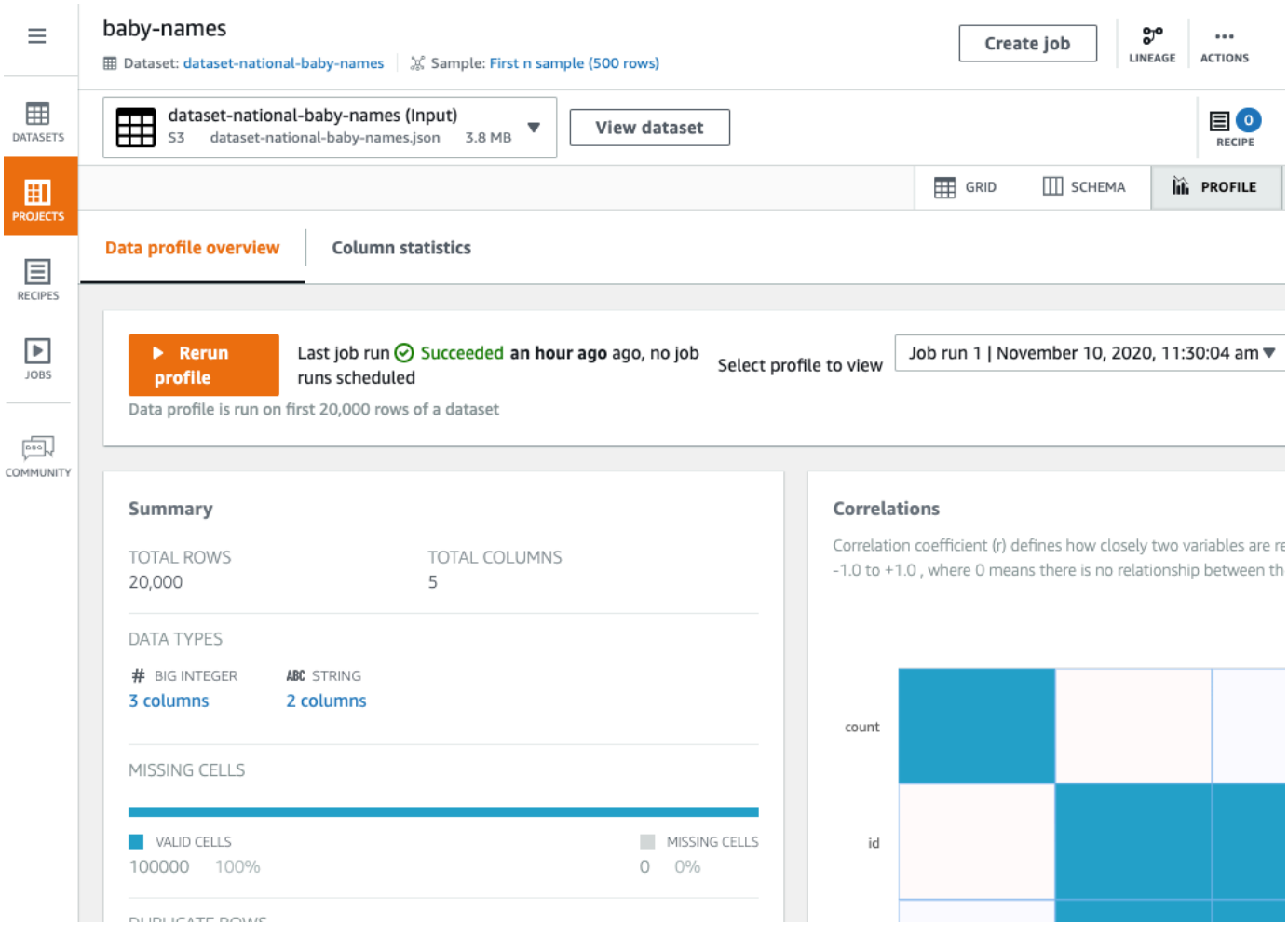
1. 選擇建立任務，然後輸入任務的名稱。
2. 針對任務輸出，選擇檔案類型的 CSV。
3. 尋找或建立AWS帳戶中要寫入 DataBrew 任務輸出的 Amazon S3 儲存貯體和資料夾：

   - 如果您已有此 Amazon S3 儲存貯體和資料夾，請選擇瀏覽並找到它們。請確定您擁有兩者的寫入許可。
   - 如果您沒有此 Amazon S3 儲存貯體和資料夾，請建立它們：

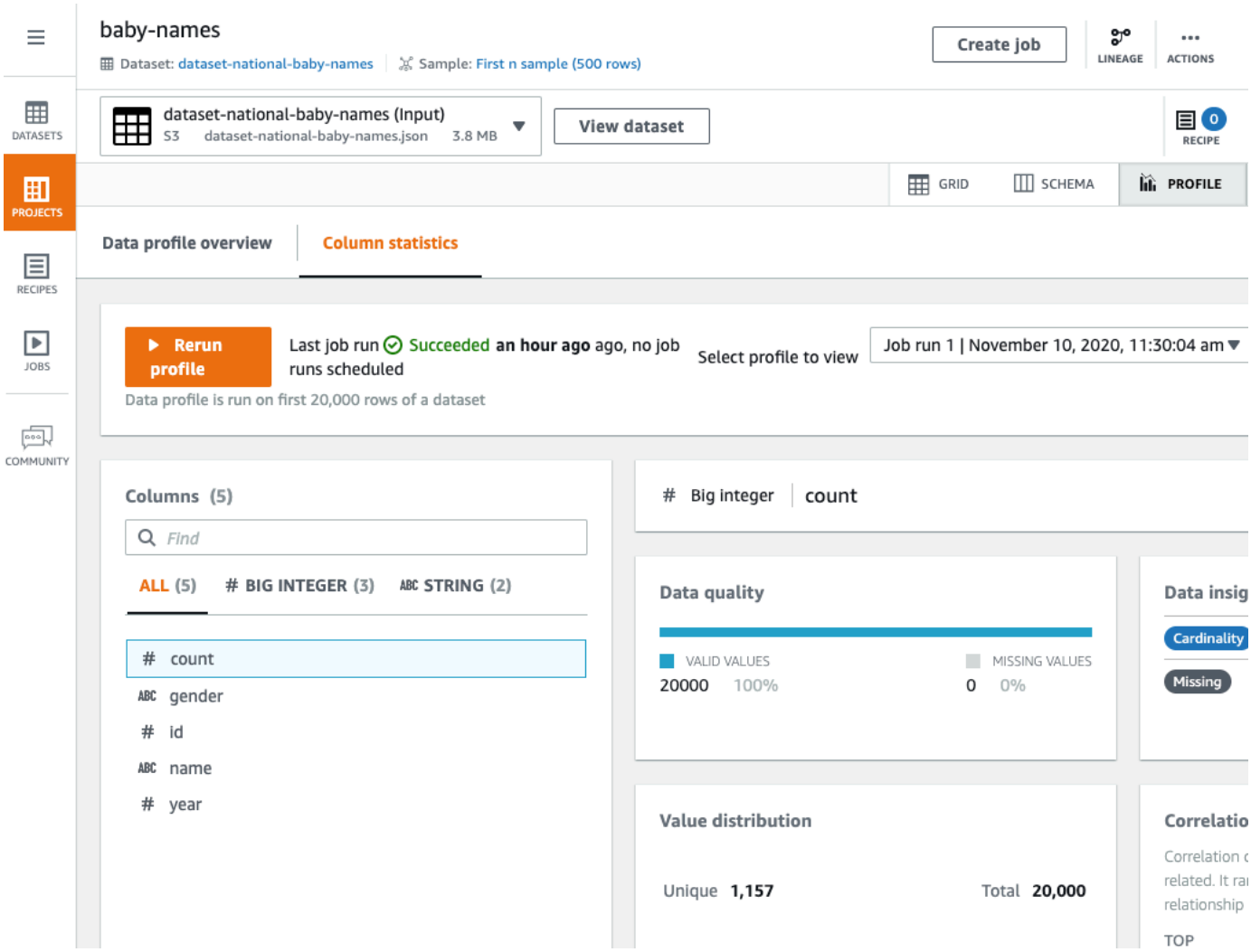
     1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
     2. 如果您沒有 Amazon S3 儲存貯體，請選擇建立儲存貯體。針對儲存貯體名稱，輸入新儲存貯體的唯一名稱。選擇建立儲存貯體。
     3. 從儲存貯體清單中，選擇您要使用的儲存貯體。
     4. 選擇 Create folder (建立資料夾)。針對資料夾名稱，輸入 `databrew-output`，然後選擇建立資料夾。
4. 針對存取許可，選擇允許 DataBrew 寫入 Amazon S3 輸出位置的 IAM 角色。

   對於AWS您的帳戶擁有的 S3 位置，您可以選擇`AwsGlueDataBrewDataAccessRole`服務受管角色。這樣做可讓 DataBrew 存取您擁有的 S3 資源。
5. 將其他設定保留為預設值，然後選擇建立和執行任務。
6. 任務執行完成之後，工作區會顯示資料設定檔的圖形摘要。

   資料設定檔概觀索引標籤會顯示資料特性的高階摘要，如下列螢幕擷取畫面所示。



資料欄統計資料索引標籤會顯示資料值的column-by-column明細：



刪除專案

如果您不再需要專案，可以將其刪除。

刪除專案

1. 在導覽窗格中，選擇專案。
2. 選擇您要刪除的專案，然後在動作中，選擇刪除。

建立和使用AWS Glue DataBrew配方

在 DataBrew 中，配方是一組資料轉換步驟。您可以將這些步驟套用至資料範例，或將相同的配方套用至資料集。

開發配方最簡單的方法是建立 DataBrew 專案，您可以在其中以互動方式使用資料範例，如需詳細資訊，請參閱 建立和使用AWS Glue DataBrew專案。在專案建立工作流程中，會建立新的（空白）配方並連接到專案。然後，您可以透過新增資料轉換開始建置配方。

> **Note**
>
> 您可以在單一 DataBrew 配方中包含最多 100 個資料轉換。

隨著您繼續開發配方，您可以透過發佈配方來儲存工作。DataBrew 會維護配方的已發佈版本清單。您可以在配方任務中使用任何已發佈的版本，以執行配方（在配方任務中）來轉換資料集。您也可以下載配方步驟的副本，以便在其他專案或其他資料集轉換中重複使用配方。

您也可以使用AWS Command Line Interface(AWS CLI) 或其中一個AWS SDKs，以程式設計方式開發 DataBrew 配方。在 DataBrew API 中，轉換稱為配方動作。

> **Note**
>
> 在互動式 DataBrew 專案工作階段中，您套用的每個資料轉換都會導致呼叫 DataBrew API。這些 API 呼叫會自動發生，您不必知道behind-the-scenes詳細資訊。

即使您不是程式設計人員，了解配方的結構以及 DataBrew 如何組織配方動作也很有幫助。

主題

- 發佈新的配方版本
- 定義配方結構

發佈新的配方版本

您可以在互動式 DataBrew 專案工作階段中發佈新版本的配方。

發佈新的配方版本

1. 在配方窗格中，選擇發佈。
2. 輸入此版本配方的描述，然後選擇發佈。

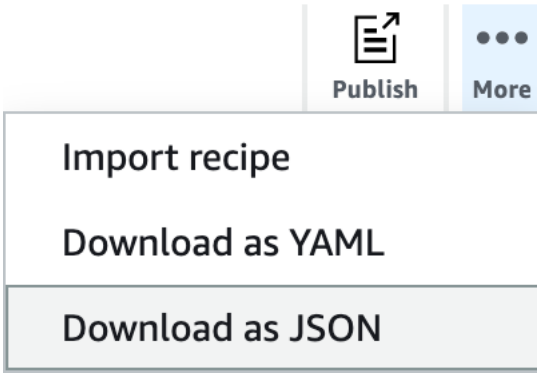
您可以從導覽窗格選擇 PROJECTS，以檢視所有已發佈的配方及其版本。

定義配方結構

當您第一次使用 DataBrew 主控台建立專案時，您可以定義與該專案相關聯的配方。如果您沒有現有的配方，主控台會為您建立一個配方。

當您在 主控台中使用專案時，您可以使用轉換工具列將動作套用至資料集的範例資料。當您繼續建置配方時，主控台會顯示配方步驟和這些步驟的順序。您可以重複和精簡配方，直到您對步驟感到滿意為止。

在 中入門AWS Glue DataBrew，您會建置配方來轉換知名 chess 遊戲的資料集。您可以選擇下載為 JSON 或下載為 YAML，以下載配方步驟的副本，如下列螢幕擷取畫面所示。



下載的 JSON 檔案包含對應至您新增至配方之轉換的配方動作。

新配方沒有任何步驟。您可以將新配方表示為空白 JSON 清單，如下所示。

```
[ ]
```

以下是 的此類檔案範例`chess-project-recipe`。JSON 清單包含數個描述配方步驟的物件。JSON 清單中的每個物件都以大括號 () 括住{ }。JSON 行以逗號分隔。

```
[
    {
        "Action": {
```

```
            "Operation": "REMOVE_VALUES",
            "Parameters": {
                "sourceColumn": "black_rating"
            }
        },
        "ConditionExpressions": [
            {
                "Condition": "LESS_THAN",
                "Value": "1800",
                "TargetColumn": "black_rating"
            }
        ]
    },
    {
        "Action": {
            "Operation": "REMOVE_VALUES",
            "Parameters": {
                "sourceColumn": "white_rating"
            }
        },
        "ConditionExpressions": [
            {
                "Condition": "LESS_THAN",
                "Value": "1800",
                "TargetColumn": "white_rating"
            }
        ]
    },
    {
        "Action": {
            "Operation": "GROUP_BY",
            "Parameters": {
                "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"winner\",
\"targetColumnName\":\"winner_count\",\"targetColumnDataType\":\"int\",\"functionName
\":\"COUNT\"}]",
                "sourceColumns": "[\"winner\",\"victory_status\"]",
                "useNewDataFrame": "true"
            }
        }
    },
    {
        "Action": {
            "Operation": "REMOVE_VALUES",
            "Parameters": {
```

```
                "sourceColumn": "winner"
            }
        },
        "ConditionExpressions": [
            {
                "Condition": "IS",
                "Value": "[\"draw\"]",
                "TargetColumn": "winner"
            }
        ]
    },
    {
        "Action": {
            "Operation": "REPLACE_TEXT",
            "Parameters": {
                "pattern": "mate",
                "sourceColumn": "victory_status",
                "value": "checkmate"
            }
        }
    },
    {
        "Action": {
            "Operation": "REPLACE_TEXT",
            "Parameters": {
                "pattern": "resign",
                "sourceColumn": "victory_status",
                "value": "other player resigned"
            }
        }
    },
    {
        "Action": {
            "Operation": "REPLACE_TEXT",
            "Parameters": {
                "pattern": "outoftime",
                "sourceColumn": "victory_status",
                "value": "ran out of time"
            }
        }
    }
]
```

如果我們只為新動作新增新的行，則更容易看到每個動作都是個別行，如下所示。

```
[
 { "Action": { "Operation": "REMOVE_VALUES", "Parameters": { "sourceColumn":
 "black_rating" } }, "ConditionExpressions": [ { "Condition": "LESS_THAN", "Value":
 "1800", "TargetColumn": "black_rating" } ] },
 { "Action": { "Operation": "REMOVE_VALUES", "Parameters": { "sourceColumn":
 "white_rating" } }, "ConditionExpressions": [ { "Condition": "LESS_THAN", "Value":
 "1800", "TargetColumn": "white_rating" } ] },
 { "Action": { "Operation": "GROUP_BY", "Parameters": { "groupByAggFunctionOptions":
 "[{\"sourceColumnName\":\"winner\",\"targetColumnName\":\"winner_count\",
\"targetColumnDataType\":\"int\",\"functionName\":\"COUNT\"}]", "sourceColumns":
 "[\"winner\",\"victory_status\"]", "useNewDataFrame": "true" } } },
 { "Action": { "Operation": "REMOVE_VALUES", "Parameters": { "sourceColumn":
 "winner" } }, "ConditionExpressions": [ { "Condition": "IS", "Value": "[\"draw\"]",
 "TargetColumn": "winner" } ] },
 { "Action": { "Operation": "REPLACE_TEXT", "Parameters": { "pattern": "mate",
 "sourceColumn": "victory_status", "value": "checkmate" } } },
 { "Action": { "Operation": "REPLACE_TEXT", "Parameters": { "pattern": "resign",
 "sourceColumn": "victory_status", "value": "other player resigned" } } },
 { "Action": { "Operation": "REPLACE_TEXT", "Parameters": { "pattern": "outoftime",
 "sourceColumn": "victory_status", "value": "ran out of time" } } }
]
```

動作會依序執行，順序與 檔案中的順序相同：

- REMOVE_VALUES – 若要篩選出玩家評分低於 1，800 的所有遊戲，A 類西洋棋玩家所需的最低評分。此動作有兩個出現次數：一個用於移除不至少 A 類玩家的黑邊玩家，另一個用於移除不在此關卡的白邊玩家。
- GROUP_BY – 摘要資料。在此情況下，GROUP_BY 會根據 winner(black 和) 的值，將資料列排序為群組white。然後，每個群組都會進一步細分，並根據 victory_status(mate、outoftime、 resign和) 的值將資料列排序為子群組draw。最後，計算每個子群組的出現次數。產生的摘要接著會取代原始資料範例。
- REMOVE_VALUES – 刪除以 結尾的遊戲結果draw。
- REPLACE_TEXT – 修改 的值victory_status。此動作有三個出現次數：mate、 resign和 各一個oufoftime。

在互動式 DataBrew 專案工作階段中，每個RecipeAction對應至您套用至資料範例的資料轉換。

DataBrew 提供超過 200 個配方動作。如需詳細資訊，請參閱配方步驟和函數參考。

使用條件

您可以使用條件來縮小配方動作的範圍。條件用於篩選資料的轉換，例如，根據特定資料欄值移除不需要的資料列。

讓我們進一步了解來自 的配方動作chess-project-recipe。

```
{
  "Action": {
    "Operation": "REMOVE_VALUES",
    "Parameters": {
      "sourceColumn": "black_rating"
    }
  },
  "ConditionExpressions": [
    {
      "Condition": "LESS_THAN",
      "Value": "1800",
      "TargetColumn": "black_rating"
    }
  ]
}
```

此轉換會讀取 black_rating 欄中的值。ConditionExpressions 清單會決定篩選條件：black_rating任何值小於 1，800 的資料列都會從資料集中移除。

配方中的後續轉換對 執行相同的動作white_rating。如此一來，資料僅限於每個玩家 （黑色或白色） 的評分為 A 級或更高等級的遊戲。

以下是套用至角色資料欄的另一個條件範例。

```
{
  "Action": {
    "Operation": "REMOVE_VALUES",
    "Parameters": {
      "sourceColumn": "winner"
    }
  },
  "ConditionExpressions": [
    {
      "Condition": "IS",
      "Value": "[\"draw\"]",
```

```
        "TargetColumn": "winner"
      }
    ]
  }
```

此轉換會讀取資料winner欄中的值，尋找值draw並移除這些資料列。如此一來，資料僅限於有明確優勝者的遊戲。

DataBrew 支援下列條件：

- IS – 欄中的值與條件中提供的值相同。
- IS_NOT – 資料欄中的值與條件中提供的值不同。
- IS_BETWEEN – 欄中的值介於 GREATER_THAN_EQUAL和 LESS_THAN_EQUAL 參數之間。
- CONTAINS – 欄中的字串值包含條件中提供的值。
- NOT_CONTAINS – 欄中的值不包含條件中提供的字元字串。
- STARTS_WITH – 欄中的值以條件中提供的字元字串開頭。
- NOT_STARTS_WITH – 資料欄中的值不會以條件中提供的字元字串開頭。
- ENDS_WITH – 資料欄中的值以條件中提供的字元字串結尾。
- NOT_ENDS_WITH – 資料欄中的值不會以條件中提供的字元字串結尾。
- LESS_THAN – 欄中的值小於條件中提供的值。
- LESS_THAN_EQUAL – 資料欄中的值小於或等於條件中提供的值。
- GREATER_THAN – 欄中的值大於條件中提供的值。
- GREATER_THAN_EQUAL – 欄中的值大於或等於條件中提供的值。
- IS_INVALID – 資料欄中的值具有不正確的資料類型。
- IS_MISSING – 欄中沒有值。

建立、執行和排程AWS Glue DataBrew任務

AWS Glue DataBrew有一個任務子系統，提供兩種用途：

1. 將資料轉換配方套用至 DataBrew 資料集。您可以使用 DataBrew 配方任務來執行此操作。
2. 分析資料集以建立資料的完整描述檔。您可以使用 DataBrew 設定檔任務來執行此操作。

主題

建立和使用AWS Glue DataBrew配方任務

使用 DataBrew 配方任務來清理和標準化 DataBrew 資料集中的資料，並將結果寫入您選擇的輸出位置。執行配方任務不會影響資料集或基礎來源資料。當任務執行時，它會以唯讀方式連線至來源資料。任務輸出會寫入您在 Amazon S3、AWS Glue Data Catalog或支援的 JDBC 資料庫中定義的輸出位置。

使用下列程序來建立 DataBrew 配方任務。

建立配方任務

1. 登入AWS 管理主控台，並在 https：//https://console.aws.amazon.com/databrew/ 開啟 DataBrew 主控台。
2. 從導覽窗格中選擇 JOBS，選擇配方任務索引標籤，然後選擇建立任務。
3. 輸入任務的名稱，然後選擇建立配方任務。
4. 針對任務輸入，輸入您要建立之任務的詳細資訊：要處理的資料集名稱，以及要使用的配方。

   配方任務使用 DataBrew 配方來轉換資料集。若要使用配方，請務必先發佈。
5. 設定您的任務輸出設定。

   為您的任務輸出提供目的地。如果您沒有為輸出目的地設定 DataBrew 連線，請先在 DATASETS 索引標籤上進行設定，如 中所述資料來源和輸出支援的連線。選擇下列其中一個輸出目的地：

   - Amazon S3，無論是否AWS Glue Data Catalog支援
   - Amazon Redshift，無論是否AWS Glue Data Catalog支援

- JDBC
- Snowflake 資料表
- AWS Glue Data Catalog支援 的 Amazon RDS 資料庫資料表。Amazon RDS 資料庫資料表支援下列資料庫引擎：
  - Amazon Aurora
  - MySQL
  - Oracle
  - PostgreSQL
  - Microsoft SQL Server
- Amazon S3 AWS Glue Data Catalog支援。

對於以 為基礎的AWS Glue Data Catalog輸出AWS Lake Formation，DataBrew 僅支援取代現有的檔案。在此方法中，會取代檔案，以保持資料存取角色的現有 Lake Formation 許可不變。此外，DataBrew 會優先考慮資料表中的 Amazon S3 位置AWS Glue Data Catalog。因此，您無法在建立配方任務時覆寫 Amazon S3 位置。

在某些情況下，任務輸出中的 Amazon S3 位置與資料目錄資料表中的 Amazon S3 位置不同。在這些情況下，DataBrew 會使用目錄資料表中的 Amazon S3 位置自動更新任務定義。當您更新或啟動現有的任務時，它會執行此操作。

6. 僅針對 Amazon S3 輸出目的地，您有進一步的選擇：

   a. 選擇 Amazon S3 的其中一個可用資料輸出格式、選用的壓縮，以及選用的自訂分隔符號。輸出檔案支援的分隔符號與輸入相同：逗號、冒號、分號、管道、標籤、插入符號、反斜線和空格。如需格式化詳細資訊，請參閱下表。

Format (格式)	副檔名 （未壓縮）	副檔名 （壓縮）
逗號分隔值	`.csv`	`.csv.snappy` ，`.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` ，`csv.br`
標籤分隔值	`.csv`	`.tsv.snappy` ，`.tsv.gz`, `.tsv.lz4`, `tsv.bz2`, `.tsv.deflate` ，`tsv.br`

Format (格式)	副檔名 （未壓縮）	副檔名 （壓縮）
Apache Parquet	.parquet	.parquet.snappy , .parquet.gz , .parquet.lz4 , .parquet.lzo , .parquet.br
AWS Glue Parquet	不支援	.glue.parquet.snappy
Apache Avro	.avro	.avro.snappy , .avro.gz, .avro.lz4 , .avro.bz2 , .avro.deflate , .avro.br
Apache ORC	.orc	.orc.snappy , .orc.lzo, .orc.zlib
XML	.xml	.xml.snappy , .xml.gz, .xml.lz4, .xml.bz2, .xml.deflate , .xml.br
JSON （僅限 JSON 行格式）	.json	.json.snappy , .json.gz, .json.lz4 , json.bz2, .json.deflate , .json.br
Tableau Hyper	不支援	不適用

b. 選擇是否輸出單一檔案或多個檔案。Amazon S3 的檔案輸出有三個選項：

- 自動產生檔案 （建議） – 讓 DataBrew 決定輸出檔案的最佳數量。
- 單一檔案輸出 – 產生單一輸出檔案。此選項可能會導致額外的任務執行時間，因為需要後置處理。

- 多個檔案輸出 – 您是否已指定任務輸出的檔案數目。有效值為 2–999。如果使用資料欄分割，或輸出中的資料列數目少於您指定的檔案數目，則輸出的檔案可能少於您指定的檔案數目。

c. （選用）選擇配方任務輸出的資料欄分割。

資料欄分割提供另一種將配方任務輸出分割成多個檔案的方式。資料欄分割可以與新的或現有的 Amazon S3 輸出或新的 Data Catalog Amazon S3 輸出搭配使用。它不能與現有的 Data Catalog Amazon S3 資料表搭配使用。輸出檔案是根據您指定的資料欄名稱值。如果您指定的資料欄名稱是唯一的，產生的 Amazon S3 資料夾路徑會根據資料欄名稱的順序而定。

如需資料欄分割的範例，請參閱資料欄分割的範例下列的 。

7. （選用）選擇啟用任務輸出的加密，以加密 DataBrew 寫入輸出位置的任務輸出，然後選擇加密方法：

- 使用 SSE-S3 加密 – 使用伺服器端加密搭配 Amazon S3 受管加密金鑰來加密輸出。
- Use AWS Key Management Service(AWS KMS) – 使用 加密輸出AWS KMS。若要使用此選項，請選擇您要使用的AWS KMS金鑰的 Amazon Resource Name (ARN)。如果您沒有AWS KMS金鑰，您可以選擇建立金鑰來建立AWS KMS金鑰。

8. 針對存取許可，選擇允許 DataBrew 寫入輸出位置的AWS Identity and Access Management(IAM)角色。對於您的帳戶AWS擁有的位置，您可以選擇`AwsGlueDataBrewDataAccessRole`服務受管角色。這樣做可讓 DataBrew 存取您擁有AWS的資源。

9. 在進階任務設定窗格中，您可以為您的任務執行方式選擇更多選項：

- 單位數量上限 – DataBrew 使用多個運算節點處理任務，並行執行。預設節點數量為 5。節點數量上限為 149。
- 任務逾時 – 如果任務需要超過您在此設定的執行分鐘數，則會失敗並出現逾時錯誤。預設值為 2，880 分鐘或 48 小時。
- 重試次數 – 如果任務在執行時失敗，DataBrew 可以嘗試再次執行。根據預設，不會重試任務。
- 為任務啟用 Amazon CloudWatch Logs – 允許 DataBrew 將診斷資訊發佈至 CloudWatch Logs。這些日誌可用於故障診斷目的，或進一步了解任務的處理方式。

10. 對於排程任務，您可以套用 DataBrew 任務排程，讓您的任務在特定時間或重複執行。如需詳細資訊，請參閱使用排程自動化任務執行。

11. 設定如您想要，請選擇建立任務。或者，如果您想要立即執行任務，請選擇建立並執行任務。

您可以在任務執行時檢查其狀態，以監控任務的進度。當任務執行完成時，狀態會變更為成功。任務輸出現在可在您選擇的輸出位置使用。

DataBrew 會儲存您的任務定義，以便您稍後可以執行相同的任務。若要重新執行任務，請從導覽窗格中選擇任務。選擇您要使用的任務，然後選擇執行任務。

資料欄分割的範例

作為資料欄分割的範例，假設您指定三個資料欄，其中每一列都包含兩個可能值之一。資料Dept欄可以有 值Admin或 Eng。資料Staff-type欄可以有 值Part-time或 Full-time。資料Location欄可以有 值Office1或 Office2。任務輸出的 Amazon S3 儲存貯體如下所示。

```
s3://bucket/output-folder/Dept=Admin/Staff-type=Part-time/Area=Office1/
jobId_timestamp_part0001.csv
s3://bucket/output-folder/Dept=Admin/Staff-type=Part-time/Location=Office2/
jobId_timestamp_part0002.csv
s3://bucket/output-folder/Dept=Admin/Staff-type=Full-time/Location=Office1/
jobId_timestamp_part0003.csv
s3://bucket/output-folder/Dept=Admin/Staff-type=Full-time/Location=Office2/
jobId_timestamp_part0004.csv
s3://bucket/output-folder/Dept=Eng/Staff-type=Part-time/Location=Office1/
jobId_timestamp_part0005.csv
s3://bucket/output-folder/Dept=Eng/Staff-type=Part-time/Location=Office2/
jobId_timestamp_part0006.csv
s3://bucket/output-folder/Dept=Eng/Staff-type=Full-time/Location=Office1/
jobId_timestamp_part0007.csv
s3://bucket/output-folder/Dept=Eng/Staff-type=Full-time/Location=Office2/
jobId_timestamp_part0008.csv
```

使用排程自動化任務執行

您可以隨時重新執行 DataBrew 任務，也可以使用排程自動化 DataBrew 任務執行。

重新執行 DataBrew 任務

1. 登入AWS 管理主控台，並在 https：//https://console.aws.amazon.com/databrew/ 開啟 DataBrew 主控台。
2. 在導覽窗格中，選擇任務。選擇您要執行的任務，然後選擇執行任務。

若要在特定時間或定期執行 DataBrew 任務，請建立 DataBrew 任務排程。然後，您可以將任務設定為根據排程執行。

建立 DataBrew 任務排程

1. 在 DataBrew 主控台的導覽窗格中，選擇任務。選擇排程索引標籤，然後選擇新增排程。
2. 輸入排程的名稱，然後選擇執行頻率的值：
   - 週期性 – 選擇您希望任務執行的頻率（例如，每 12 小時）。然後選擇要在哪一天執行任務。或者，您可以輸入任務執行的時間。
   - 在特定時間 – 輸入您希望任務執行的時間。然後選擇要在哪一天執行任務。
   - 輸入 CRON – 輸入有效的 Cron 表達式來定義任務排程。如需詳細資訊，請參閱使用配方任務的 cron 表達式。
3. 當您滿意設定後，請選擇 Save (儲存)。

將任務與排程建立關聯

1. 在導覽窗格中，選擇任務。
2. 選擇您要使用的任務，然後針對動作選擇編輯。
3. 在排程任務窗格中，選擇關聯排程。選擇您要使用的排程名稱。
4. 當您滿意設定後，請選擇 Save (儲存)。

使用配方任務的 cron 表達式

Cron 表達式有六個必要欄位，以空格隔開。語法如下。

```
Minutes Hours Day-of-month Month Day-of-week Year
```

在上述語法中，下列值和萬用字元用於指定的欄位。

欄位	Values (數值)	萬用字元
分鐘	0–59	, - * /
小時	0–23	, - * /
月中的日	1–31	, - * ? / L W
月	1-12 或 JAN-DEC	, - * /

欄位	Values (數值)	萬用字元
週中的日	1-7 或 SUN-SAT	, - * ? / L
年	1970-2199	, - * /

請依照下列方式使用這些萬用字元：

- , (逗號) 萬用字元包含額外的值。在 Month欄位中， JAN,FEB,MAR包含 1 月、2 月和 3 月。
- - (en dash) 萬用字元指定範圍。在 Day 欄位中，1–15 包含指定月份的第 1 天到第 15 天。
- * (星號) 包含欄位中所有的值。在 Hours欄位中，* 包含每小時。
- / (斜線) 萬用字元用於指定增量。在 Minutes欄位中，您可以輸入 **1/10**以指定每 10 分鐘，從小時的第一分鐘開始 （例如，第 11、第 21 和第 31 分鐘）。
- ? (問號) 萬用字元用於表示不限定任何一個。例如，假設在 Day-of-month 欄位中輸入 7。如果您不在乎一週的第七天是哪一天，則可以輸入 ？ 在 Day-of-week 欄位中。
- Day-of-month 或 Day-of-week 欄位中的 L 萬用字元指定當月或當週的最後一天。
- W 萬用字元在 Day-of-month 欄位可指定任務日。在 Day-of-month 欄位，3W 指定的是月份中最接近第三個任務日的日子。

這些欄位和值有下列限制：

- 您無法在同一個 cron 表達式中指定 Day-of-month 和 Day-of-week 欄位。如果您在其中一個欄位指定了數值，就必須在另一個欄位中使用 ? (問號)。
- 不支援導致速率超過 5 分鐘的 Cron 表達式。

建立排程時，您可以使用下列 cron 字串範例。

分鐘	小時	月中的日	月	週中的日	年	意義
0	10	*	*	?	*	在每天上午 10：00 (UTC) 執行

分鐘	小時	月中的日	月	週中的日	年	意義
15	12	*	*	?	*	在每天下午 12:15 (UTC) 執行
0	18	?	*	MON-FRI	*	在每週一至週五下午 6:00 (UTC) 執行
0	8	1	*	?	*	每個月第一天的上午 8：00 (UTC) 執行
0/15	*	*	*	?	*	每 15 分鐘執行
0/10	*	?	*	MON-FRI	*	在週一至週五每 10 分鐘執行
0/5	8-17	?	*	MON-FRI	*	在週一至週五上午 8:00 至下午 5:55 (UTC) 之間每 5 分鐘執行

例如，您可以使用下列 cron 表達式，在 UTC 每天 12：15 執行任務。

```
15 12 * * ? *
```

刪除任務和任務排程

如果您不再需要任務或任務排程，可以將其刪除。

若要刪除工作

1. 在導覽窗格中，選擇任務。
2. 選擇您要刪除的任務，然後針對動作選擇刪除。

刪除任務排程

1. 在導覽窗格中，選擇任務，然後選擇排程索引標籤。
2. 選擇您要刪除的排程，然後在動作中，選擇刪除。

建立和使用AWS Glue DataBrew設定檔任務

設定檔任務會對資料集執行一系列評估，並將結果輸出至 Amazon S3。資料分析收集的資訊可協助您了解資料集，並決定您可能想要在配方任務中執行的資料準備步驟。

執行設定檔任務最簡單的方法是使用預設 DataBrew 設定。您可以在執行設定檔任務之前對其進行設定，以便只傳回您想要的資訊。

使用下列程序來建立 DataBrew 設定檔任務。

建立設定檔任務

1. 登入AWS 管理主控台並開啟 DataBrew 主控台，網址為 https：//https://console.aws.amazon.com/databrew/。
2. 從導覽窗格中選擇 JOBS，選擇設定檔任務索引標籤，然後選擇建立任務。
3. 輸入任務的名稱，然後選擇建立設定檔任務。
4. 針對任務輸入，提供要描述的資料集名稱。
5. （選用）在資料設定檔組態窗格中設定下列項目：

   - 資料集層級組態 – 為資料集中的所有資料欄設定設定檔任務的詳細資訊。

     或者，您可以開啟偵測和計數資料集中重複資料列的功能。您也可以選擇啟用相互關聯矩陣，然後選取資料欄，以查看多個資料欄中的值關聯程度。如需您可以在資料集層級設定的統計資料詳細資訊，請參閱 資料集層級的可設定統計資料。您可以在 DataBrew 主控台或使用 DataBrew API 或AWS SDKs設定統計資料。
   - 資料欄層級組態 – 使用預設設定檔組態設定，您可以選取要包含在設定檔任務中的資料欄。使用新增組態覆寫來選取要限制所收集統計資料數量的資料欄，或覆寫特定統計資料的預設組態。

如需您可以在資料欄層級設定的統計資料詳細資訊，請參閱 資料欄層級的可設定統計資料。您可以在 DataBrew 主控台或使用 DataBrew API 或AWS SDKs設定統計資料。

請確定您指定的任何組態覆寫都適用於您在設定檔任務中包含的資料欄。如果您為資料欄設定的不同覆寫之間存在衝突，則最後一個衝突覆寫具有優先順序。

6. （選用）您可以建立資料品質規則，並套用與此資料集相關聯的其他規則集，或移除已套用的規則集。如需資料品質驗證的詳細資訊，請參閱 在 中驗證資料品質AWS Glue DataBrew。
7. 在進階任務設定窗格中，您可以為您的任務執行方式選擇更多選項：
   - 單位數量上限 – DataBrew 使用多個運算節點處理任務，並行執行。預設節點數量為 5。節點數量上限為 149。
   - 任務逾時 – 如果任務需要超過您在此設定的執行分鐘數，則會失敗並出現逾時錯誤。預設值為 2，880 分鐘或 48 小時。
   - 重試次數 – 如果任務在執行時失敗，DataBrew 可以嘗試再次執行。根據預設，不會重試任務。
   - 為任務啟用 Amazon CloudWatch Logs – 允許 DataBrew 將診斷資訊發佈至 CloudWatch Logs。這些日誌可用於故障診斷目的，或進一步了解任務的處理方式。
8. 對於關聯的排程，您可以套用 DataBrew 任務排程，讓您的任務在特定時間或重複執行。如需詳細資訊，請參閱使用排程自動化任務執行。
9. 設定如您想要，請選擇建立任務。或者，如果您想要立即執行任務，請選擇建立並執行任務。

在 中以程式設計方式建置設定檔任務組態AWS Glue DataBrew

在本節中，您可以找到描述檔任務步驟和函數的描述，以程式設計方式使用。您可以從AWS Command Line Interface(AWS CLI) 使用它們，或使用其中一個AWS SDKs。

在設定檔任務中，您可以自訂組態，以控制 DataBrew 如何評估您的資料集。您可以將組態套用至資料集，或將其套用至特定資料欄。您可以在建立設定檔任務時建置組態，然後隨時更新。

設定檔組態結構包含四個部分：

- ProfileColumns 區段
- DatasetStatisticsConfiguration 區段
- ColumnStatisticsConfigurations 區段
- 用於設定 PII 的 EntityDetectorConfiguration 區段

以下是範例。

```
{
    "ProfileColumns": [
        {
            "Name": "example"
        },
        {
            "Regex": "example.*"
        }
    ],
    "DatasetStatisticsConfiguration": {
        "IncludedStatistics": [
            "CORRELATION"
        ],
        "Overrides": [
            {
                "Statistic": "CORRELATION",
                "Parameters": {
                    "columnSelectors": "[{\"name\":\"example\"}, {\"regex\":\"example.*
\"}]"
                }
            }
        ]
    },
    "ColumnStatisticsConfigurations": [
        {
            "Selectors": [
                {
                    "Name": "example"
                }
            ],
            "Statistics": {
                "IncludedStatistics": [
                    "CORRELATION",
                    "DUPLICATE_ROWS_COUNT"
                ],
                "Overrides": [
                    {
                        "Statistic": "VALUE_DISTRIBUTION",
                        "Parameters": {
                            "binNumber": "10"
                        }
                    }
                ]
```

```
            }
        }
    ]
}
```

ProfileColumns 區段

在結構的 ProfileColumns區段中，從資料集設定您要在設定檔任務中評估的資料欄。ProfileColumns是資料欄選取器 (Selectors) 的清單。您可以在資料欄選取器中指定資料欄名稱或規則表達式。範例如下。

```
"ProfileColumns": [{"Name": "example"}, {"Regex": "example.*"}]
```

指定 ProfileColumns 時，只有名稱符合 中名稱或規則表達式的資料欄ProfileColumns才會包含在設定檔任務中。如果設定檔任務不支援所選資料欄的資料類型，DataBrew 會在任務執行期間略過選取的資料欄。

如果 ProfileColumns 未定義，設定檔任務會評估所有支援的資料欄。支援的欄是包含所支援資料類型資料的資料欄：ByteType、ShortType、IntegerType、LongType、FloatType、String、DoubleType或 Boolean。

DatasetStatisticsConfiguration 區段

在 結構的 DatasetStatisticsConfiguration區段中，您可以建置資料欄間評估的組態。組態包含 IncludedStatistics和 Overrides。範例如下。

```
"DatasetStatisticsConfiguration": {
    "IncludedStatistics": ["CORRELATION"],
    "Overrides": [
        {
            "Statistic": "CORRELATION",
            "Parameters": {
                "columnSelectors": "[{\"name\":\"example\"}, {\"regex\":\"example.*
\"}]"
            }
        }
    ]
}
```

您可以將評估名稱新增至 ，以選取您想要擁有的評估IncludedStatistics。範例如下。

```
"IncludedStatistics": ["CORRELATION", "DUPLICATE_ROWS_COUNT"]
```

當您指定 時IncludedStatistics，設定檔任務中只會包含清單中的評估。如果IncludedStatistics未定義，則設定檔任務會使用預設設定執行所有支援的評估。您可以將 NONE 新增至 以排除所有評估IncludedStatistics。範例如下。

```
"IncludedStatistics": ["NONE"]
```

資料集層級的可設定統計資料

在您結構的 DatasetStatisticsConfiguration區段中，設定檔任務支援下表所示的評估。

統計資料名稱	Description	支援的資料類型	預設狀態	設定檔結果的屬性	設定檔結果的類型
DUPLICATE_ROWS_COUNT	資料集中重複資料列的計數	全部	Enable	duplicateRowsCoun	Int
相互關聯	兩欄之間的 Pearson 相關性係數	number	Enable	相互關聯（在每個選取的欄中）	物件

在 中IncludedStatistics，您可以透過新增覆寫來覆寫每個評估的預設設定。每個覆寫都包含特定評估的名稱和參數映射。

在 中DatasetStatisticsConfiguration，設定檔任務支援CORRELATION覆寫。此覆寫會從所選資料欄清單中計算兩個資料欄之間的 Pearson 相關性係數。預設設定是選取前 10 個數字資料欄。您可以指定欄數或欄選取器清單，以覆寫預設設定。

CORRELATION 採用這些參數：

- columnNumber – 數值資料欄的數量。設定檔任務會從資料集中選取前 n 個資料欄。此值應大於 1。使用 "ALL" 選取所有數值欄。

- columnSelectors: – 欄選取器的清單。每個選取器都可以有欄名稱或規則表達式。

範例如下。

```
{
    "Statistic": "CORRELATION",
    "Parameters": {
        "columnSelectors": "[{\"name\":\"example\"}, {\"regex\":\"example.*\"}]"
    }
}
```

ColumnStatisticsConfigurations 區段

在 結構的 ColumnStatisticsConfigurations區段中，您可以針對特定資料欄建置組態。 ColumnStatisticsConfigurations是ColumnStatisticsConfiguration設定清單。在 中ColumnStatisticsConfiguration，有 Selectors、資料欄選取器清單，以及Statistics用於統計資料組態的 。範例如下。

```
{
    "Selectors": [{"Name": "example"}
    ],
    "Statistics": {
       "IncludedStatistics": ["CORRELATION", "DUPLICATE_ROWS_COUNT"]
        "Overrides": [
            {
                "Statistic": "VALUE_DISTRIBUTION",
                "Parameters": {
                    "binNumber": "10"
                }
            }
        ]
    }
}
```

Selectors 是欄選取器的清單。如同 ProfileColumns，您可以在每個資料欄選擇器中指定資料欄名稱或規則表達式。當您指定 時Selectors，資料欄組態會套用至符合 中任何資料欄選擇器的資料欄Selectors。否則，組態會套用至所有支援的欄。

在 中Statistics，您可以覆寫所選資料欄的設定。如同 DatasetStatisticsConfiguration，Statistics具有 IncludedStatistics和 Overrides。

若要選取您想要的評估，請將評估名稱新增至 IncludedStatistics。

```
"IncludedStatistics": ["CORRELATION", "DUPLICATE_ROWS_COUNT"]
```

當您指定 時IncludedStatistics，設定檔任務中只會包含清單中的評估。否則，設定檔任務會使用預設設定執行所有支援的評估。

您可以將 新增至 NONE以排除所有評估IncludedStatistics。

```
"IncludedStatistics": ["NONE"]
```

在某些情況下， 中可能有多個組態ColumnStatisticsConfigurations具有不同的IncludedStatistics組態，您可以套用至相同的資料欄。在這些情況下，設定檔任務會挑選 中的最後一個組態，ColumnStatisticsConfigurations並將其套用至IncludedStatistics選取的資料欄。新的組態會覆寫較舊的組態。

資料欄層級的可設定統計資料

在 中ColumnStatisticsConfigurations，設定檔任務支援下表所示的評估。

此資料表number中支援的資料類型 表示屬性的資料類型為下列其中一項：ByteType、ShortType、IntegerTypeLongType、FloatType、 或 DoubleType。

統計資料名稱	Description	支援的資料類型	預設狀態	設定檔結果的屬性	設定檔結果的類型
–	欄的名稱。	全部	–	name	string
–	資料欄的資料類型。	全部	–	type	string
DISTINCT_VALUES_COUNT	不同值的數量。不同的值是至少出現一次的值。	number/boolean/string	已啟用	distinctValuesCount	Int
熵	Entropy （資訊理論）。	number/boolean/string	已啟用	熵	Double

統計資料名稱	Description	支援的資料類型	預設狀態	設定檔結果的屬性	設定檔結果的類型
INTER_QUARTILE_RANGE	範圍介於數字的第 25% 到第 75% 之間。	number	已啟用	interquartileRange	Double
KURTOSIS	資料欄的峰度。	number	已啟用	峰度	Double
MAX	資料欄中的最大值。	number/string 長度	已啟用	max	Int/Double
MAXIMUM_VALUES	資料欄中的最大值清單及其計數。	number	已啟用	maximumVlues	清單
MEAN	資料欄中值的平均值。	number/string 長度	已啟用	mean	Double
MEDIAN	資料欄中值的中位數。	number/string 長度	已啟用	median	Double
MEDIAN_ABSOLUTE_DEVIATION	每個資料點與數值資料欄中位數之間的絕對差異中位數。	number	已啟用	medianAbsoluteDeviation	Double
MIN	資料欄中的最小值。	number/string 長度	已啟用	min	Int/Double
MINIMUM_VALUES	資料欄中的最小值及其計數的清單。	number	已啟用	minimumVlues	清單
MISSING_VALUES_COUNT	資料欄中遺失值的數量。Null 和空白字串視為遺失。	全部	已啟用	missingValuesCount	Int
MODE	資料欄中最常出現的值。如果經常出現數個值，則模式是其中一個值。	number/string 長度	已啟用	模式	Int/Double

統計資料名稱	Description	支援的資料類型	預設狀態	設定檔結果的屬性	設定檔結果的類型
MOST_COMMON_VALUES	欄中最常見的值清單。	number/boolean/string	已啟用	mostCommnValues	清單
OUTLIER_DETECTION	透過 Z_score 演算法偵測欄中的極端值。計算極端值的數量，並從偵測到的極端值擷取範例清單。	number/string長度	已啟用	zScoreOutliersCount、zScoreOutliersSample	Int/List
百分位數	數值欄的百分位數值 (5%、25%、75%、95%)。	number	已啟用	百分位數 5、百分位數 25、百分位數 75、百分位數 95	Double
RANGE	資料欄中的值範圍。	number	已啟用	range	Int/Double
偏斜	資料欄中值的偏斜。	number	已啟用	偏斜	Double
STANDARD_DEVIATION	資料欄中值的無偏差樣本標準差。	number/string長度	已啟用	standardDeviation	Double
SUM	資料欄中的值總和。	number	已啟用	sum	Int/Double
UNIQUE_VALUES_COUNT	唯一值的數量。唯一值表示該值僅顯示一次。	number/boolean/string	已啟用	uniqueValuesCount	Int
VALUE_DISTRIBUTION	依範圍測量資料欄中值的分佈。	number/string長度	已啟用	valueDistribution	清單
VARIANCE	資料欄中值的差異。	number	已啟用	variance	Double

統計資料名稱	Description	支援的資料類型	預設狀態	設定檔結果的屬性	設定檔結果的類型
Z_SCORE_DISTRIBUTION	依範圍測量資料點 z 分數的分佈。	number	已啟用	zScoreDistribution	清單
ZEROS_COUNT	資料欄中的零 (0) 數目。	number	已啟用	zerosCount	Int

在 中IncludedStatistics，您可以透過新增覆寫來覆寫每個評估的預設參數。每個覆寫都包含特定評估的名稱和參數映射。

ColumnStatisticsConfigurations 欄的參數

在 中ColumnStatisticsConfigurations，設定檔任務支援下列參數。

在某些情況下， 中可能有多個組態ColumnStatisticsConfigurations具有不同的IncludedStatistics組態，您可以套用至相同的資料欄。在這些情況下，設定檔任務會挑選 中的最後一個組態，ColumnStatisticsConfigurations並將其套用至IncludedStatistics選取的資料欄。新的組態會覆寫較舊的組態。

MAXIMUM_VALUES

列出數值欄中的最大值及其計數。預設清單大小為 5。您可以指定 的值來覆寫清單大小sampleSize。

設定

sampleSize – 清單的大小，包含數值欄中值的最大數量和計數。此值應大於 0。使用 "ALL" 列出所有值。

範例

```
{
    "Statistic": "MAXIMUM_VALUES",
    "Parameters": {
        "sampleSize": "5"
    }
```

```
}
```

MINIMUM_VALUES

列出數值欄中的最小值及其計數。預設清單大小為 5。您可以指定 的值來覆寫清單大小sampleSize。

設定

sampleSize – 清單的大小，包含數值欄中值的最大數量和計數。此值應大於 0。使用 "ALL" 列出所有值。

範例

```
{
    "Statistic": "MINIMUM_VALUES",
    "Parameters": {
        "sampleSize": "5"
    }
}
```

MOST_COMMON_VALUES

列出資料欄中最常見的值及其計數。預設清單大小為 50。您可以指定 的值來覆寫清單大小sampleSize。

設定

sampleSize – 清單的大小，包含數值欄中值的最大數量和計數。此值應大於 0。使用 "ALL" 列出所有值。

範例

```
{
    "Statistic": "MOST_COMMON_VALUES",
    "Parameters": {
        "sampleSize": "50"
    }
```

```
}
```

OUTLIER_DETECTION

依 Z_score 演算法偵測數值欄或字串欄中的極端值 （根據字串長度）。

您的設定檔任務會計算極端值數量，並產生極端值及其 z 分數的範例清單。範例清單會依 z-score 的絕對值排序。預設清單大小為 50。

當 Z_Score 演算法偏離平均值超過標準差閾值時，會將值識別為極端值。預設極端值閾值為 3。

您可以提供另一個閾值，即輕度閾值，以取得更多資訊。您的輕度閾值應小於閾值。此功能預設為關閉。指定輕微閾值時，您的設定檔任務會再傳回一個計數 zScoreMildOutliersCount。此外，在這種情況下， zScoreOutliersSample可以包含輕度閾值極端值的範例。

設定

- threshold – 偵測極端值時要使用的閾值。此值應該大於或等於 0。
- mildThreshold – 偵測極端值時要使用的輕微閾值。此值應該大於或等於 0 且小於 threshold。
- sampleSize – 包含欄中極端值的清單大小。使用 "ALL" 列出所有值。

範例

```
{
    "Statistic": "OUTLIER_DETECTION",
    "Parameters": {
        "threshold": "5",
        "mildThreshold": "3.5",
        "sampleSize": "20"
    }
}
```

VALUE_DISTRIBUTION

依值的範圍測量資料欄中值的分佈。設定檔任務會依數值範圍將數值欄或字串欄 （根據字串長度） 中的值分組到 bin，並產生 bin 清單。儲存貯體是連續的，而儲存貯體的上限是下一個儲存貯體的下限。

設定

binNumber – 儲存貯體數量。此值應大於 0。

範例

```
{
    "Statistic": "VALUE_DISTRIBUTION",
    "Parameters": {
        "binNumber": "5"
    }
}
```

Z_SCORE_DISTRIBUTION

測量數值欄中值 z 分數的分佈。設定檔任務會依數值範圍將值的 z 分數分組到 bin，並產生 bin 清單。儲存貯體是連續的，而儲存貯體的上限是下一個儲存貯體的下限。

設定

binNumber – 儲存貯體數量。此值應大於 0。

範例

```
{
    "Statistic": "Z_SCORE_DISTRIBUTION",
    "Parameters": {
        "binNumber": "5"
    }
}
```

用於設定 PII 的 EntityDetectorConfiguration 區段

在 結構的 EntityDetectorConfiguration區段中，您可以設定資料集中的實體類型，讓 DataBrew 偵測為設定檔任務的個人身分識別資訊 (PII)。

EntityTypes

您可以設定希望 DataBrew 偵測為設定檔任務 PII 的實體類型。當 EntityDetectorConfiguration 未定義時，實體偵測會停用。您可以在資料集中偵測到下列實體類型：

- USA_SSN
- EMAIL
- USA_ITIN
- USA_PASSPORT_NUMBER
- PHONE_NUMBER
- USA_DRIVING_LICENSE
- BANK_ACCOUNT
- CREDIT_CARD
- IP_ADDRESS
- MAC_ADDRESS
- USA_DEA_NUMBER
- USA_HCPCS_CODE
- USA_NATIONAL_PROVIDER_IDENTIFIER
- USA_NATIONAL_DRUG_CODE
- USA_HEALTH_INSURANCE_CLAIM_NUMBER
- USA_MEDICARE_BENEFICIARY_IDENTIFIER
- USA_CPT_CODE
- PERSON_NAME
- DATE

USA_ALL 也支援實體類型群組，並包含上述所有實體類型，但 PERSON_NAME和 除外DATE。

的類型EntityTypes是字串的陣列。

AllowedStatistics

設定允許在包含偵測到實體的資料欄上執行的統計資料。如果 AllowedStatistics未定義，則不會在包含偵測到實體的資料欄上計算統計資料。資料欄層級的可設定統計資料 如需 AllowedStatistics 參數的有效值清單，請參閱 。

的類型AllowedStatistics是 AllowedStatistics 物件的陣列。

中的安全性AWS Glue DataBrew

的雲端安全性AWS是最高優先順序。身為AWS客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是AWS與您之間的共同責任。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 –AWS負責保護在 Cloud AWS中執行AWS服務的基礎設施。AWS也為您提供可安全使用的服務。在AWS合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用的合規計劃AWS Glue DataBrew，請參閱AWS合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用AWS的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 時套用共同責任模型AWS Glue DataBrew。下列主題說明如何設定 DataBrew 以符合您的安全與合規目標。您也會了解如何使用其他AWS服務來協助您監控和保護 DataBrew 資源。

主題

中的資料保護AWS Glue DataBrew

DataBrew 提供多種功能，旨在協助保護您的資料。

主題

- 識別和處理個人身分識別資訊 (PII)
- DataBrew 對其他服務的相依性AWS

將AWS共同責任模式應用於AWS Glue DataBrew中的資料保護。如此模型所述，AWS負責保護執行所有 的全域基礎設施AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。 如需歐洲資料保護的詳細資訊，請參閱一般資料保護規則 (GDPR) 中心。

基於資料保護目的，我們建議您保護AWS 帳戶登入資料，並使用AWS IAM Identity Center或AWS Identity and Access Management(IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與AWS資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取AWS活動的資訊，請參閱AWS CloudTrail《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用AWS加密解決方案，以及其中的所有預設安全控制AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在AWS透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 DataBrew 或使用主控台、API AWS CLI、AWS SDKs的其他 AWS 服務時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

靜態加密

DataBrew 支援 DataBrew 專案和任務的靜態資料加密。專案和任務可以讀取加密的資料，而任務可以呼叫 AWS Key Management Service(AWS KMS) 來寫入加密的資料，以產生金鑰和解密資料。您也可以使用 KMS 金鑰來加密 DataBrew 任務所產生的任務日誌。您可以使用 DataBrew 主控台或 DataBrew API 來指定加密金鑰。

> **Important**
>
> AWS Glue DataBrew僅支援對稱AWS KMS 金鑰。如需詳細資訊，請參閱《 AWS Key Management Service開發人員指南》中的 AWS KMS 金鑰。

當您在已啟用加密的 DataBrew 中建立任務時，您可以使用 DataBrew 主控台來指定 S3-managed伺服器端加密金鑰 (SSE-S3) 或存放在 (SSE-KMS) 中的AWS KMS KMS 金鑰，以加密靜態資料。

> **Important**
>
> 當您使用 Amazon Redshift 資料集時，卸載至所提供暫存目錄的物件會使用 SSE-S3 加密。

加密 DataBrew 任務寫入的資料

DataBrew 任務可以寫入加密的 Amazon S3 目標和加密的 Amazon CloudWatch Logs。

主題

- 設定 DataBrew 以使用加密
- 為 VPC AWS KMS任務建立路由至
- 使用AWS KMS 金鑰設定加密

設定 DataBrew 以使用加密

請依照此程序設定 DataBrew 環境以使用加密。

設定 DataBrew 環境以使用加密

1. 建立或更新AWS KMS 金鑰，以將AWS KMS許可授予傳遞給 DataBrew 任務的AWS Identity and Access Management(IAM) 角色。這些 IAM 角色用於加密 CloudWatch Logs 和 Amazon S3 目標。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的使用AWS KMS加密 CloudWatch Logs 中的日誌資料。

   在下列範例中，*"role1"*、 *"role2"*和 *"role3"*是傳遞給 DataBrew 任務的 IAM 角色。此政策陳述式描述一個 KMS 金鑰政策，授予列出的 IAM 角色使用此 KMS 金鑰加密和解密的許可。

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logs.region.amazonaws.com",
        "AWS": [
            "role1",
            "role2",
            "role3"
        ]
    },
    "Action": [
        "kms:Encrypt*",
        "kms:Decrypt*",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:Describe*"
    ],
    "Resource": "*"
}
```

`Service` 陳述式 (顯示為 `"Service": "logs.region.amazonaws.com"`) 在您使用金鑰加密 CloudWatch Logs 時為必要項目。

2. 在使用AWS KMS金鑰ENABLED之前，請確定金鑰設定為 。

如需使用AWS KMS金鑰政策指定許可的詳細資訊，請參閱在 中使用金鑰政策AWS KMS。

為 VPC AWS KMS任務建立路由至

您可以透過在 Virtual Private Cloud (VPC) 內的私有端點直接連接至AWS KMS，而無需連接至網際網路。當您使用 VPC 端點時，VPC 與 之間的通訊AWS KMS會完全在AWS網路中執行。

您可以在AWS KMS VPC 內建立 VPC 端點。如果沒有此步驟，您的 DataBrew 任務可能會因為 而失敗`kms timeout`。如需詳細說明，請參閱《 AWS Key Management Service開發人員指南》中的AWS KMS透過 VPC 端點連線至 。

當您遵循這些指示時，在 VPC 主控台上，請務必執行下列動作：

- 選擇啟用私有 DNS 名稱。
- 針對安全群組，選擇您用於存取 Java Database Connectivity (JDBC) 之 DataBrew 任務的安全群組（包括自我參考規則）。

當您執行存取 JDBC 資料存放區的 DataBrew 任務時，DataBrew 必須具有端點的路由AWS KMS。您可以使用網路位址轉譯 (NAT) 閘道或AWS KMS VPC 端點來提供路由。若要建立 NAT 閘道，請參閱 Amazon VPC 使用者指南中的 NAT 閘道。

使用AWS KMS 金鑰設定加密

當您在任務上啟用加密時，它會同時套用到 Amazon S3 和 CloudWatch。傳遞的 IAM 角色必須具有下列AWS KMS許可。

如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的下列主題：

- 如需關於 SSE-S3 的詳細資訊，請參閱使用伺服器端加密與 Amazon S3 受管加密金鑰 (SSE-S3) 保護資料。
- 如需 的相關資訊SSE-KMS，請參閱使用伺服器端加密搭配AWS KMS 受管金鑰 (SSE-KMS) 保護資料。

傳輸中加密

AWS為傳輸中的資料提供 Secure Sockets Layer (SSL) 加密。

DataBrew 支援 JDBC 資料來源AWS Glue。連線至 JDBC 資料來源時，DataBrew 會在您的AWS Glue連線上使用設定，包括需要 SSL 連線選項。如需詳細資訊，請參閱《 AWS Glue開發人員指南》中的AWS Glue連線屬性 -AWS Glue。

AWS KMS為 DataBrew 擷取、轉換、載入 (ETL) 處理和 提供「自有金鑰」加密和伺服器端加密AWS Glue Data Catalog。

金鑰管理

您可以使用 IAM 搭配 DataBrew 來定義使用者、AWS資源、群組、角色，以及有關存取、拒絕等的精細政策。

您可以根據組織的需求，使用資源型和身分型政策來定義中繼資料的存取權。資源類型政策會列出允許或拒絕存取您資源的委託人，讓您可以設定像是跨帳戶存取等政策。身分政策則是特別連接到 IAM 內使用者、群組和角色的政策。

DataBrew 支援建立您自己的AWS KMS key「使用您自己的金鑰」加密。DataBrew 也為 DataBrew 任務提供使用 之 KMS 金鑰AWS KMS的伺服器端加密。

識別和處理個人身分識別資訊 (PII)

當您建置分析函數或機器學習模型時，您需要保護措施，以防止個人身分識別資訊 (PII) 資料的公開。PII 是個人資料，可用於識別個人，例如地址、銀行帳戶號碼或電話號碼。例如，當資料分析師和資料科學家使用資料集來探索一般人口統計資訊時，他們不應該存取特定個人的 PII。

DataBrew 提供資料遮罩機制，以在資料準備過程中混淆 PII 資料。根據您的組織需求，有不同的 PII 資料修訂機制可用。您可以混淆 PII 資料，讓使用者無法將其還原，也可以讓混淆恢復。

在 DataBrew 中識別和遮罩 PII 資料涉及建置一組轉換，供客戶用來修訂 PII 資料。此程序的一部分是在 DataBrew 主控台的資料設定檔概觀儀表板中提供 PII 資料偵測和統計資料。

您可以使用下列資料遮罩技術：

- 替代 - 將 PII 資料取代為其他外觀真實的值。
- 隨機播放 – 從不同資料列中的相同資料欄隨機播放值。
- 確定性加密 – 將確定性加密演算法套用至資料欄值。確定性加密一律會為值產生相同的加密文字。
- 機率加密 – 將機率加密演算法套用至資料欄值。每次套用機率加密都會產生不同的加密文字。
- 解密 – 根據加密金鑰解密資料欄。
- 剔除或刪除 – 將特定欄位取代為 null 值或刪除資料欄。
- 遮罩 – 使用角色雜湊或遮罩欄中的某些部分。
- 雜湊：將雜湊函數套用至資料欄值。

如需使用轉換的詳細資訊，請參閱個人身分識別資訊 (PII) 配方步驟。如需使用設定檔任務來偵測 PII 的詳細資訊，包括可偵測到的實體類型清單，請參閱 EntityDetectorConfiguration 一節，以程式設計方式在建置設定檔任務組態中設定 PII。

DataBrew 對其他服務的相依性AWS

若要使用 DataBrew 主控台，您需要一組最低許可，才能使用AWS您帳戶的 DataBrew 資源。除了這些 DataBrew 許可之外，主控台還需要下列服務的許可：

- CloudWatch Logs 顯示日誌的許可。
- 列出和傳遞角色的 IAM 許可。
- 列出 VPCs、子網路、安全群組、執行個體和其他物件的 Amazon EC2 許可。DataBrew 會在執行 DataBrew 任務時，使用這些許可來設定 Amazon EC2 項目，例如 VPCs。

- 列出儲存貯體和物件的 Amazon S3 許可。
- AWS Glue讀取AWS Glue結構描述物件的許可，例如資料庫、分割區、資料表和連線。
- AWS Lake Formation使用 Lake Formation 資料湖的許可。

的身分和存取管理AWS Glue DataBrew

AWS Identity and Access Management(IAM) 是AWS 服務，可協助管理員安全地控制對AWS資源的存取。IAM 管理員可控制誰可以進行身分驗證（登入）和授權（具有許可）來使用 DataBrew 資源。IAM 是您可以免費使用AWS 服務的 。

主題

使用身分驗證

身分驗證是您AWS使用身分憑證登入 的方式。您必須以AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如AWS IAM Identity Center(IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入使用者指南》中的如何登入您的AWS 帳戶。

對於程式設計存取，AWS提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS第 4 版簽署程序。

AWS 帳戶根使用者

建立 時AWS 帳戶，您會從一個名為AWS 帳戶theroot 使用者的登入身分開始，該身分可完整存取所有 AWS 服務和 資源。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的要求人類使用者使用聯合身分提供者來AWS使用臨時憑證存取 。

IAM 群組會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過從使用者切換到 IAM 角色 （主控台） 或呼叫AWS CLI或AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源AWS來控制AWS中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時AWS， 會評估這些政策。大多數政策會以 JSON 文件AWS的形式存放在 中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的 JSON 政策概觀。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的在受管政策與內嵌政策之間選擇。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。您必須在資源型政策中指定主體。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的AWS受管政策。

DataBrew 不支援以資源為基礎的政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱《Amazon Simple Storage Service 開發人員指南》中的存取控制清單 (ACL) 概觀。

DataBrew 不支援 ACLs。

其他政策類型

AWS支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《 IAM 使用者指南》中的 IAM 實體許可界限。
- 服務控制政策 (SCP) — 為AWS Organizations中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations使用者指南》中的資源控制政策 (RCP)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參《IAM 使用者指南》中的工作階段政策。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何AWS在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS Glue DataBrew而且AWS Lake Formation

AWS Glue DataBrew支援AWS Glue Data Catalog資料表的AWS Lake Formation許可。當資料集使用向 Lake Formation 註冊的AWS Glue Data Catalog資料表時，提供給專案或任務的 IAM 角色必須在資料表上具有 DESCRIBE 和 SELECT Lake Formation 許可。

AWS Glue DataBrew支援根據 寫入AWS Glue Data Catalog資料表AWS Lake Formation。當 DataBrew 任務使用向 Lake Formation 註冊的 Data Catalog 時，提供給任務的 IAM 角色必須具有 Lake Formation 中涉及的資料表的 INSERT、ALTER 和 DELETE 許可。IAM 角色必須具有與 Data Catalog 資料表相關聯的資料位置的`glue:UpdateTable`許可和許可。

AWS Glue DataBrew如何使用 IAM

在使用 IAM 管理 DataBrew 的存取權之前，您應該了解哪些 IAM 功能可與 DataBrew 搭配使用。若要全面了解 DataBrew 和其他AWS服務如何與 IAM 搭配使用，請參閱《IAM 使用者指南》中的與 IAM AWS搭配使用的 服務。

主題

- DataBrew 身分型政策
- DataBrew 中的資源型政策
- DataBrew IAM 角色

DataBrew 身分型政策

透過 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，還有在何種條件下允許或拒絕動作。DataBrew 支援特定動作、資源和條件索引鍵。若要了解您在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的 JSON 政策元素參考。

動作

管理員可以使用AWS JSON 政策來指定誰可以存取內容。也就是說，AWS JSON 政策可以指定哪些主體可以對哪些資源執行動作，以及在哪些條件下執行動作。

JSON 政策的動作元素說明您可以在政策中允許或拒絕存取的動作。政策動作的名稱通常會和相關聯的 AWS API 作業相同。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

DataBrew 中的政策動作在動作之前使用下列字首：databrew:。例如，若要授予某人使用 Amazon EC2 RunInstances API 作業來執行 Amazon EC2 執行個體的許可，請在其政策中加入 ec2:RunInstances 動作。政策陳述式必須包含 Action 或 NotAction 元素。DataBrew 會定義自己的一組動作，描述您可以執行的任務。

若要在單一陳述式中指定多個 動作，請用逗號分隔，如下所示。

```
"Action": [
      "databrew:CreateRecipeJob",
      "databrew:UpdateSchedule"
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，如需指定開頭是 Describe 文字的所有動作，請包含以下動作：

```
"Action": "databrew:Describe*"
```

若要查看 DataBrew 動作清單，請參閱《IAM 使用者指南》中的 定義的動作AWS Glue DataBrew。

Resources

管理員可以使用AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

以下是不支援資源層級許可的 DataBrew APIs：

- ListDatasets
- ListJobs
- ListProjects
- ListRecipes
- ListRulesets
- ListSchedules

DataBrew 資料集資源具有下列 Amazon Resource Name (ARN)。

```
arn:${Partition}:databrew:${Region}:${Account}:dataset/${Name}
```

如需 ARNs 格式的詳細資訊，請參閱 Amazon Resource Name (ARNs)AWS和服務命名空間。

例如，若要在陳述式中指定i-1234567890abcdef0執行個體，請使用下列 ARN。

```
"Resource": "arn:aws:databrew:us-east-1:123456789012:dataset/my-chess-dataset"
```

如需指定屬於特定帳戶的所有執行個體，請使用萬用字元 (*)。

```
"Resource": "arn:aws:databrew:us-east-1:123456789012:dataset/*"
```

您無法在特定資源上執行一些 DataBrew 動作，例如用於建立資源的動作。在這些情況下，您必須使用萬用字元 (*)。

```
"Resource": "*"
```

若要查看 DataBrew 資源類型及其 ARNs的清單，請參閱《IAM 使用者指南》中的 定義的資源AWS Glue DataBrew。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS Glue DataBrew定義的動作。

條件索引鍵

DataBrew 不提供任何服務特定的條件金鑰，但支援使用一些全域條件金鑰。若要查看所有AWS全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS全域條件內容索引鍵。

範例

若要檢視 DataBrew 身分型政策的範例，請參閱 的身分型政策範例AWS Glue DataBrew。

DataBrew 中的資源型政策

DataBrew 不支援以資源為基礎的政策。

DataBrew IAM 角色

IAM 角色是您AWS帳戶中具有特定許可的實體。

搭配 DataBrew 使用臨時登入資料

您可以搭配聯合使用暫時憑證、擔任 IAM 角色，或是擔任跨帳戶角色。您可以透過呼叫 AssumeRole 或 GetFederationToken 等AWS STS API 操作來取得臨時安全登入資料。

DataBrew 支援使用臨時登入資料。

服務連結角色

服務連結角色可讓AWS服務存取其他服務中的資源，以代表您完成 動作。服務連結角色會顯示在您的 IAM 帳戶中，並由該服務所擁有。 管理員可以檢視，但不能編輯服務連結角色的許可。

在 DataBrew 中選擇 IAM 角色

當您在 DataBrew 中建立資料集資源時，您可以選擇 IAM 角色以允許 DataBrew 代表您存取。如果您先前已建立服務角色或服務連結角色，則 DataBrew 會為您提供可供選擇的角色清單。請務必視需要選擇允許對 Amazon S3 儲存貯體或AWS Glue Data Catalog資源進行讀取存取的角色。

的身分型政策範例AWS Glue DataBrew

根據預設，使用者和角色沒有建立或修改 DataBrew 資源的許可。他們也無法使用AWS 管理主控台 AWS CLI或AWS APIs執行任務。管理員必須建立 IAM 政策，授與使用者和角色在指定資源上執行特定 API 操作所需的許可。管理員接著必須將這些政策連接至需要這些許可的使用者或群組。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱 IAM 使用者指南中的在 JSON 索引標籤上建立政策。

主題

- 政策最佳實務
- 使用 DataBrew 主控台
- 允許使用者檢視自己的許可
- 根據標籤管理 DataBrew 資源

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 DataBrew 資源。這些動作可能會讓您的AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用AWS受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS受管政策。它們可在您的 中使用AWS 帳戶。我們建議您定義

特定於使用案例AWS的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS受管政策或任務職能的AWS受管政策。

- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 等使用服務動作AWS服務，您也可以使用條件來授予其存取權CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

使用 DataBrew 主控台

若要存取AWS Glue DataBrew主控台，您必須擁有一組最低許可。這些許可必須可讓您列出和檢視AWS帳戶中 DataBrew 資源的詳細資訊。如果您建立比最低必要許可更嚴格的身分型政策，主控台對於具有該政策的使用者或角色不會如預期運作。

為了確保使用者和角色可以使用 DataBrew 主控台，請將下列AWS受管政策連接至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

```
AWSDataBrewConsoleAccess
```

對於僅呼叫AWS CLI或 DataBrew API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合您嘗試執行之 API 作業的動作就可以了。

允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或AWS CLIAWS API 以程式設計方式完成此動作的許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

根據標籤管理 DataBrew 資源

您可以在身分型政策中使用條件，根據標籤管理 DataBrew 資源，例如刪除、更新或描述資源。下列範例顯示拒絕刪除專案的政策。不過，只有在專案標籤擁有者具有 admin 的值時，才會拒絕刪除。此政策也會授予在主控台上拒絕此動作所需的許可。

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "DeleteResourceInConsole",
         "Effect": "Allow",
         "Action": "databrew:DeleteProject",
         "Resource": "*"
       },
       {
         "Sid": "DenyDeleteProjectIfAdminTag",
         "Effect": "Deny",
         "Action": "databrew:DeleteProject",
         "Resource": "arn:aws:databrew:*:*:project/*",
         "Condition": {
            "StringEquals": {"aws:ResourceTag/Owner": "admin"}
         }
      }
   ]
}
```

您可以將此政策連接到您帳戶中的 使用者。如果名為 richard-roe 的使用者嘗試刪除 DataBrew 專案，則資源不得標記 Owner=admin 或 owner=admin。否則，拒絕使用者刪除專案的許可。條件標籤金鑰擁有者同時符合擁有者和擁有者，因為條件金鑰名稱不區分大小寫。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

> **Note**
>
> ListDatasets、ListJobs、ListProjects、ListRecipes、ListRulesets 和 ListSchedules 不支援標籤型存取控制。

AWS的 受管政策AWS Glue DataBrew

若要新增許可給使用者、群組和角色，使用AWS受管政策比自行撰寫政策更容易。建立 IAM 客戶受管政策需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的AWS受管政策。這些政策涵蓋常見的使用案例，並且可在您的帳戶中使用AWS。如需AWS受管政策的詳細資訊，請參閱《IAM 使用者指南》中的 AWS受管政策。

AWS服務會維護和更新AWS受管政策。您無法變更AWS受管政策中的許可。服務偶爾會將其他許可新增至AWS受管政策，以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和

角色)。當新功能啟動或新操作可用時，服務最有可能更新AWS受管政策。服務不會從AWS受管政策中移除許可，因此政策更新不會破壞您現有的許可。

此外，AWS支援跨多個 服務之任務函數的受管政策。例如，ReadOnlyAccessAWS受管政策提供所有AWS服務和資源的唯讀存取權。當服務啟動新功能時， 會為新操作和資源AWS新增唯讀許可。如需任務函數政策的清單和說明，請參閱《IAM 使用者指南》中的AWS任務函數的受管政策。

AWS受管政策的 DataBrew 更新

檢視自此服務開始追蹤這些變更以來DataBrew 受AWS管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 DataBrew 文件歷史記錄頁面上的 RSS 摘要。您可以在 的AWS IAM 主控台中找到 受管政策AwsGlueDataBrewFullAccessPolicy。

變更	描述	日期
AWSGlueDataBrewServiceRole –AWS Glue已新增 的讀取許可。	此更新新增了 `glue:GetCustomEntityType` 。執行已啟用 PII 身分的AWS Glue DataBrew設定檔任務需要此許可。	2024 年 3 月 20 日
AWSGlueDataBrewServiceRole -AWS Glue已新增 的讀取許可。	此更新新增了 `glue:BatchGetCustomEntityTypes` 。執行已啟用 PII 身分的AWS Glue DataBrew設定檔任務需要此許可。	2022 年 5 月 9 日
AwsGlueDataBrewFullAccessPolicy - 已新增 Amazon Redshift-Data DescribeStatements 和 Amazon S3 GetLifecycleConfiguration 的讀取許可。	此更新新增 `redshift-data:DescribeStatement` 以支援在建立以 Amazon Redshift 為基礎的資料集時驗證您的 SQL。它也會新增 `s3:GetLifecycleConfiguration` ，以評估您作為暫時目錄提供的 Amazon S3 儲存貯體字首是否已設定生命週期。此外，此變更會以包含所有 DataBrew API 的明確許	2022 年 2 月 4 日

變更	描述	日期
	可清單取代「databrew：*」許可。 APIs	
AwsGlueDataBrewFullAccessPolicy - 已新增AWS Secrets Manager 的讀取/寫入許可。	此更新`secretsmanager:GetSecretValue` 會為名為 的秘密新增 `secretsmanager:CreateSecret` 和 `databrew!default` ，這是與 DataBrew 轉換搭配使用的預設秘密。此外，它會為字首為 的秘密新增 CreateSecret 許可`AwsGlueDataBrew-` ，以便從 DataBrew 主控台建立秘密。GenerateRandom，如 AWS Key Management Service API 參考中所述，用於產生加密安全的隨機位元組字串。	2021 年 11 月 18 日
AWSGlueDataBrewServiceRole - 已新增AWS Secrets Manager 的讀取/寫入許可。	此更新`secretsmanager:GetSecretValue` 會為名為 的秘密新增 `databrew!default` ，這是與 DataBrew 轉換搭配使用的預設秘密。	2021 年 11 月 18 日

變更	描述	日期
AwsGlueDataBrewFullAccessPolicy - 已新增AWS Secrets Manager 的讀取/寫入許可。	此更新secretsmanager:GetSecretValue 會為名為 的秘密新增 secretsmanager:CreateSecret 和 databrew!default ，這是與 DataBrew 轉換搭配使用的預設秘密。此外，它會為字首為 CreateSecret 的秘密新增許可AwsGlueDataBrew- ，以便從 DataBrew 主控台建立秘密。 kms:GenerateRandom (https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html) 用於產生加密安全的隨機位元組字串。	2021 年 11 月 18 日
AWSGlueDataBrewServiceRole - 已新增AWS Secrets Manager 的讀取/寫入許可。	此更新secretsmanager:GetSecretValue 會為名為 的秘密新增 databrew!default ，這是與 DataBrew 轉換搭配使用的預設秘密。	2021 年 11 月 18 日
AwsGlueDataBrewFullAccessPolicy - 已新增AWS Glue目錄資料庫的讀取許可和建立AWS Glue目錄資料表的許可。	此更新新增列出AWS Glue目錄資料庫的許可，並在現有資料庫下建立新的目錄資料表，做為設定 DataBrew 任務輸出的一部分。	2021 年 6 月 30 日

變更	描述	日期
AwsGlueDataBrewFullAccessPolicy - 已新增 Amazon AppFlow 資料集功能的讀取/寫入許可。	此更新新增讀取現有 Amazon AppFlow 流程和流程執行以及建立流程執行的許可。	2021 年 4 月 28 日
AwsGlueDataBrewFullAccessPolicy - 已新增資料庫資料集的讀取許可。	此更新新增許可，以讀取現有AWS Glue連線並建立新的AWS Glue連線，以搭配 DataBrew 使用。 此外，為了讓主控台更輕鬆地建立新連線，它允許列出 Amazon VPC 資源和 Amazon Redshift 叢集。它也提供列出 的許可，但不要讀取AWS Secrets Manager秘密。	2021 年 3 月 30 日
DataBrew 開始追蹤變更	DataBrew 開始追蹤其AWS受管政策的變更。	2021 年 3 月 30 日

對 中的身分和存取權進行故障診斷AWS Glue DataBrew

使用以下資訊來協助您診斷和修正使用 DataBrew 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 DataBrew 中執行動作
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許AWS帳戶外的人員存取我的 DataBrew 資源

我無權在 DataBrew 中執行動作

如果AWS 管理主控台告訴您無權執行 動作，請聯絡您的管理員尋求協助。您的管理員是為您提供簽署憑證的人員。

mateojackson 使用者嘗試使用主控台檢視專案的詳細資訊，但卻沒有 databrew:DescribeProject 許可時，會出現以下範例錯誤。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 databrew:DescribeProject on resource: my-example-project
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-project* 動作存取 databrew:*GetProject* 資源。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 iam:PassRole動作，您的政策必須更新，以允許您將角色傳遞給 DataBrew。

有些AWS 服務可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 的 IAM marymajor 使用者嘗試使用主控台在 DataBrew 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的AWS管理員。您的管理員提供您的簽署憑證。

我想要允許AWS帳戶外的人員存取我的 DataBrew 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 DataBrew 是否支援這些功能，請參閱 AWS Glue DataBrew如何使用 IAM。
- 若要了解如何AWS 帳戶在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶的另一個 中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方AWS 帳戶擁有。

- 如需了解如何透過聯合身分提供存取權，請參閱《IAM 使用者指南》中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

在 DataBrew 中記錄和監控

監控是維護 DataBrew 和AWS解決方案可靠性、可用性和效能的重要部分。您應該從AWS解決方案的所有部分收集監控資料，以便在發生多點失敗時更輕鬆地偵錯。AWS提供數種工具來監控 DataBrew 資源並回應潛在事件：

Amazon CloudWatch 警示

使用 Amazon CloudWatch 警示，您可在自己指定的一段時間內監看單一指標。如果指標超過指定的閾值，則會傳送通知至 Amazon SNS 主題或AWS Auto Scaling政策。CloudWatch 警示不會因為它們處於特定狀態而叫用動作。必須是狀態已變更並維持了所指定的時間長度，才會呼叫動作。

AWS CloudTrail日誌

CloudTrail 提供由使用者、角色或 DataBrew 中的AWS服務所採取之動作的記錄。您可以使用 CloudTrail 所收集的資訊，判斷對 DataBrew 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

的合規驗證AWS Glue DataBrew

在多個合規計畫AWS Glue DataBrew中，第三方稽核人員會評估 的安全與AWS合規。這些計劃包括 SOC、PCI、FedRAMP、HIPAA 等等。

若要了解 是否AWS 服務在特定合規計劃範圍內，請參閱AWS 服務合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS合規計劃。

您可以使用 下載第三方稽核報告AWS Artifact。如需詳細資訊，請參閱在 中下載報告AWS Artifact。

您使用 時的合規責任AWS 服務取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊AWS 服務，請參閱 AWS安全文件。

中的彈性AWS Glue DataBrew

AWS全球基礎設施是以AWS區域和可用區域為基礎建置的。AWS區域提供多個實體隔離和隔離的可用區域，這些區域以低延遲、高輸送量和高度備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

對於AWS Glue DataBrew，我們建議您將任務設定為使用一或多個重試。任務的重試次數是在 DataBrew 主控台的進階任務設定下設定。

如需AWS區域和可用區域的詳細資訊，請參閱 AWS全球基礎設施。

中的基礎設施安全AWS Glue DataBrew

作為受管服務的一部分，AWS Glue DataBrew受到 Amazon Web Services：安全程序概觀白皮書中所述的AWS全球網路安全程序的保護。

您可以使用AWS發佈的 API 呼叫，透過網路存取 DataBrew。用戶端必須支援 Transport Layer Security (TLS) 1.0 或更新版本。建議使用 TLS 1.2 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

主題

- AWS Glue DataBrew搭配 VPC 使用
- AWS Glue DataBrew搭配 VPC 端點使用

AWS Glue DataBrew搭配 VPC 使用

如果您使用 Amazon VPC 託管AWS資源，您可以設定AWS Glue DataBrew以根據 Amazon VPC 服務，透過虛擬私有雲端 (VPC) 路由流量。DataBrew 會先在您指定的子網路中佈建彈性網路介面來執行此操作。DataBrew 接著會將您指定的安全群組連接到該網路界面，以控制存取。指定的安全群組必須具有所有流量的自我參考傳入和傳出規則。此外，您的 VPC 必須開啟 DNS 主機名稱和解析。如需詳細資訊，請參閱《 AWS Glue開發人員指南》中的設定 VPC 以連線至 JDBC 資料存放區。

對於AWS Glue Data Catalog資料集，當您在 Data Catalog 中建立連線時，AWS Glue會設定 VPC 資訊。若要為此連線建立 Data Catalog 資料表，請從AWS Glue主控台執行爬蟲程式。如需詳細資訊，請參閱《 開發人員指南》中的填入AWS Glue Data Catalog 。 AWS Glue

對於資料庫資料集，當您從 DataBrew 主控台建立連線時，請指定您的 VPC 資訊。

若要在沒有 NAT 的情況下AWS Glue DataBrew搭配 VPC 子網路使用 ，您必須擁有閘道 VPC 端點至 Amazon S3，以及AWS Glue介面的 VPC 端點。如需詳細資訊，請參閱 Amazon VPC 文件中的建立閘道端點和介面 VPC 端點 ()。 AWS PrivateLinkDataBrew 佈建的彈性界面沒有公有 IPv4 地址，因此不支援使用 VPC 網際網路閘道。

目前不支援 Amazon S3 介面端點。如果您使用AWS Secrets Manager來存放秘密，則需要通往 Secrets Manager 的路由。如果您使用加密，則需要路由到AWS Key Management Service(AWS KMS)。

AWS Glue DataBrew搭配 VPC 端點使用

如果您使用 Amazon VPC 託管AWS資源，您可以透過佈建 VPC 端點，在 VPC 和 DataBrew 之間建立私有連線。使用此 VPC 端點，您可以進行 DataBrew API 呼叫。

DataBrew VPC 端點不需要搭配 VPC 使用 DataBrew。如需詳細資訊，請參閱AWS Glue DataBrew搭配 VPC 使用。

您可以在支援AWS Glue和 VPC 端點的所有AWS區域中AWS Glue，搭配使用 與 VPC 端點。

如需詳細資訊，請參閱 Amazon VPC 使用者指南 中的下列主題：

- 什麼是 Amazon VPC？
- 建立界面端點

中的組態和漏洞分析AWS Glue DataBrew

組態和 IT 控制是客戶AWS與您之間的共同責任。如需詳細資訊，請參閱AWS共同的責任模型。

監控AWS Glue DataBrew

監控是維護AWS Glue DataBrew和其他AWS解決方案的可靠性、可用性和效能的重要部分。AWS提供下列監控工具來監看 DataBrew、在發生錯誤時回報，以及適時採取自動動作：

- Amazon CloudWatchAWS會即時監控您的AWS資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon CloudWatch Events 可讓您為 DataBrew 中的特定事件設定自動通知。來自 DataBrew 的事件會以近乎即時的方式交付至 CloudWatch Events。您可以設定 CloudWatch Events 來監控事件並叫用目標，以回應指出資源共享變更的事件。資源共用的變更會觸發資源共用擁有者和獲得資源共用存取權的委託人的事件。如需詳細資訊，請參閱 Amazon CloudWatch Events 使用者指南。
- Amazon CloudWatch Logs 可讓您監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 及其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。
- AWS CloudTrail 會擷取由您的帳戶發出或代表AWS您的帳戶發出的 API 呼叫和相關事件。其接著會將日誌檔案交付到您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱「AWS CloudTrail使用者指南」。

主題

使用 Amazon CloudWatch 監控 DataBrew

您可以使用 CloudWatch 監控 DataBrew，這會收集原始資料並將其處理為可讀且近乎即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

AWS Glue DataBrew在 AWS/DataBrew 命名空間中報告下列指標。

指標	說明
SessionCount	客戶帳戶中的 DataBrew 工作階段總數 有效維度：LogGroupName 有效統計資訊：總和 單位：Count

使用 CloudWatch Events 自動化 DataBrew

Amazon CloudWatch Events 可讓您自動化您的AWS服務，並自動回應系統事件，例如應用程式可用性問題或資源變更。來自AWS服務的事件會以近乎即時的方式交付至 CloudWatch Events。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。可以自動觸發的動作如下：

- 叫用 Amazon EC2 執行命令
- 將事件轉傳至 Amazon Kinesis Data Streams
- 啟用AWS Step Functions狀態機器
- 通知 Amazon SNS 主題或 Amazon SQS 佇列

DataBrew 會在您AWS帳戶中的資源狀態變更時，向 CloudWatch Events 報告事件。盡可能發出事件。

以下是數個事件的範例，顯示 DataBrew 任務的各種狀態：SUCCEEDED、TIMEOUT、 FAILED和 STOPPED。

```
{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-09-07T18:57:21Z",
```

```
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
    "severity": "INFO",
    "state": "SUCCEEDED",
    "jobRunId": "db_abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789",
    "message": "Job run succeeded"
  }
}

{
  "version": "0",
  "id": "abcdef01-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-09-07T06:02:03Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
    "severity": "ERROR",
    "state": "FAILED",
    "jobRunId": "db_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
    "message": "AnalysisException: 'Path does not exist: s3://MyBucket/MyFile;'"
  }
}

{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-11-20T20:22:06Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
    "severity": "WARN",
    "state": "TIMEOUT",
    "jobRunId": "db_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
    "message": "Job run timed out"
```

```
  }
}

{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-11-20T20:22:06Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
    "severity": "INFO",
    "state": "STOPPED",
    "jobRunId": "db_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
    "message": "Job run stopped"
  }
}
```

如需詳細資訊，請參閱 Amazon CloudWatch Events 使用者指南。

使用 CloudWatch Logs 監控 DataBrew

您可以使用 CloudWatch Logs 監控 DataBrew 任務，這會從 DataBrew 任務子系統收集詳細資訊，並使其可供檢閱。如果您想要深入了解您的設定檔和配方任務正在使用的資源，或進行故障診斷，這些日誌會很有幫助，如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。

使用 記錄 DataBrew API 呼叫AWS CloudTrail

DataBrew 已與 服務整合AWS CloudTrail，此服務提供由使用者、角色或 DataBrew 中的AWS服務所採取之動作的記錄。CloudTrail 會將 DataBrew 的所有 API 呼叫擷取為事件。擷取的呼叫包括來自 DataBrew 主控台的呼叫，以及對 DataBrew API 操作的程式碼呼叫。如果您建立追蹤，則可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 DataBrew 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 所收集的資訊，判斷對 DataBrew 提出的請求。您還可以判斷提出請求的來源 IP 地址、提出請求的人員和時間以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail使用者指南」。

CloudTrail 中的 DataBrew 資訊

當您建立AWS帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 DataBrew 中發生時，該活動會與事件歷史記錄中的其他服務AWS事件一起記錄在 CloudTrail 事件中。您可以在AWS帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱《AWS CloudTrail使用者指南》中的使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您AWS帳戶中的事件，包括 DataBrew 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台中建立線索時，線索會套用至所有AWS區域。線索會記錄AWS分割區中所有區域的事件，並將日誌檔案傳送到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他AWS服務，以進一步分析和處理 CloudTrail 日誌中收集的事件資料。如需詳細資訊，請參閱《AWS CloudTrail使用者指南》 中的下列主題：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

CloudTrail 會記錄所有 DataBrew 動作，並記錄在 API 參考中。例如，對 `CreateDataset`、`UpdateRecipe` 和 `StartJobRun` 動作發出的呼叫會在 CloudTrail 記錄檔案中產生專案。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他AWS服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 DataBrew 日誌檔案項目

同樣地，CloudTrail 追蹤是一種組態，可讓您將事件做為日誌檔案交付至您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔案並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範 `CreateProfileJob` 操作：

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::1234567890:user/joe",
        "accountId": "1234567890",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "joe"
    },
    "eventTime": "2020-11-09T18:54:44Z",
    "eventSource": "databrew.amazonaws.com",
    "eventName": "CreateProfileJob",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "requestParameters": {
        "OutputLocation": {
            "Bucket": "bucketName",
            "Key": "keyName"
        },
        "DatasetName": "my-chess-dataset",
        "RoleArn": "arn:aws:iam::1234567890:role/custom-role",
        "Name": "my-profile-job"
    },
    "responseElements": {
        "Name": "my-profile-job"
    },
    "requestID": "993bc3b8-3980-48dd-961e-c1c8529eb248",
    "eventID": "f8128dfa-df29-458b-a2d5-34805b46eefd",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "1234567890"
}
```

搭配AWS Glue Databrew 使用AWS使用者通知

您可以使用AWS使用者通知來設定交付管道，以取得AWS Glue有關 Databrew 事件的通知。當事件符合您指定的規則時，便會收到通知。您可以透過多個管道接收事件通知，包括電子郵件、聊天應用程式中的 Amazon Q Developer 聊天通知或 AWS Console Mobile Application 推送通知。您也可以在主控台通知中心查看通知。AWS使用者通知支援彙總，可減少您在特定事件期間收到的通知數量。

配方步驟和函數參考

在此參考中，您可以從AWS CLI或使用其中一個AWS SDKs，以程式設計方式找到配方步驟和函數的說明。在 DataBrew 中，配方步驟是一種動作，可將原始資料轉換為已準備好供資料管道使用的形式。DataBrew 函數是一種特殊配方步驟，可根據參數執行運算。

UI 中轉換的類別包括下列項目：

- 基本資料欄配方步驟
  - 篩選條件
  - 資料行
- 資料清理配方步驟
  - 格式
  - 全新
  - 擷取
- 資料品質配方步驟
  - 缺少
  - 無效
  - Duplicates (複製)
  - 極端值
- 個人身分識別資訊 (PII) 配方步驟
  - 遮罩個人資訊
  - 取代個人資訊
  - 加密個人資訊
  - 隨機顯示資料列
- 資料欄結構配方步驟
  - Split
  - Merge
  - 建立
- 資料欄格式化配方步驟
  - 小數精確度
  - 千個分隔符號

    - 縮寫數字
- 資料結構配方步驟
    - Nest-Unnest
    - Pivot (樞紐)
    - Group
    - Join
    - UNION
- 資料科學配方步驟
    - 文字
    - 擴展
    - 映射
    - 編碼
- 函數
    - 數學函式
    - 彙總函數
    - 文字函數
    - 日期和時間函數
    - 範圍函數
    - Web 函數
    - 其他 函數

如需如何在配方中使用這些配方步驟和函數的詳細資訊（包括使用條件表達式），請參閱 定義配方結構。

下列各節說明配方步驟和函數，依其執行方式整理。

主題

- 基本資料欄配方步驟
- 資料清理配方步驟
- 資料品質配方步驟
- 個人身分識別資訊 (PII) 配方步驟
- 極端值偵測和處理配方步驟

- 資料欄結構配方步驟
- 資料欄格式化配方步驟
- 資料結構配方步驟
- 資料科學配方步驟
- 數學函式
- 彙總函數
- 文字函數
- 日期和時間函數
- 範圍函數
- Web 函數
- 其他 函數

基本資料欄配方步驟

使用這些基本資料欄配方動作，對資料執行簡單的轉換。

主題

- CHANGE_DATA_TYPE
- DELETE
- 重複
- JSON_TO_STRUCTS
- MOVE_AFTER
- MOVE_BEFORE
- MOVE_TO_END
- MOVE_TO_INDEX
- MOVE_TO_START
- RENAME
- SORT
- TO_BOOLEAN_COLUMN
- TO_DOUBLE_COLUMN

- TO_NUMBER_COLUMN
- TO_STRING_COLUMN

CHANGE_DATA_TYPE

變更現有資料欄的資料類型。

如果資料欄值無法轉換為新類型，則會以 NULL 取代。當字串資料欄轉換為整數資料欄時，可能會發生這種情況。例如，字串 "123" 會變成整數 123，但字串 "ABC" 無法變成數字，因此會以 NULL 值取代。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 新類型的資料欄。目前支援下列資料類型：
  - 位元組：1 位元組帶正負號整數。數字的範圍是 -128 到 127。
  - 短：2 位元組帶正負號整數。數字的範圍是 -32768 到 32767。
  - int：4 位元組帶正負號整數。數字的範圍是從 -2147483648 到 2147483647。
  - 長：8 位元組帶正負號整數。數字的範圍是 -9223372036854775808 到 9223372036854775807。
  - float：4 位元組單精度浮點數。
  - 雙：8 位元組雙精度浮點數。
  - 小數：符號小數，總計最多 38 位數，小數點後最多 18 位數。
  - string：字元字串值。
  - 布林值：布林值類型有兩個可能的值之一：`true` 和 `false` 或 `yes` 和 `no`。
  - 時間戳記：包含欄位年、月、日、小時、分鐘和秒的值。
  - date：包含欄位年份、月份和日期的值。

Example範例

```
{
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
```

```
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
}
```

DELETE

從資料集移除資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DELETE",
        "Parameters": {
            "sourceColumn": "extra_data"
        }
    }
}
```

重複

建立具有不同名稱但具有所有相同資料的新資料欄。舊欄和新欄都會保留在資料集中。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 重複資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
```

```
        "Operation": "DUPLICATE",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "copy_of_last_name"
        }
    }
}
```

JSON_TO_STRUCTS

將 JSON 字串轉換為靜態類型的結構。在轉換期間，它會偵測每個 JSON 物件的結構描述並將其合併，以取得最通用的結構描述來代表整個 JSON 字串。“unnestLevel” 參數指定要轉換為結構的 JSON 物件層級。

Parameters

- `sourceColumns` – 來源資料欄的清單。
- `regexColumnSelector` – 選取資料欄的規則表達式。
- `removeSourceColumn` – 布林值。如果 `true` 移除來源資料欄，否則請保留它。
- `unnestLevel` – 要取消巢狀化的關卡數量。
- `conditionExpressions` – 條件表達式。

Example範例

```
{
    "RecipeAction": {
        "Operation": "JSON_TO_STRUCTS",
        "Parameters": {
            "sourceColumns": "[\"address\"]",
            "removeSourceColumn": "true",
            "unnestLevel": "2"
       }
    }
}
```

MOVE_AFTER

將資料欄立即移至另一個資料欄之後的位置。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 另一個資料欄的名稱。指定的資料欄sourceColumn會在 指定的資料欄之後立即移動targetColumn。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOVE_AFTER",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "height_cm"
        }
    }
}
```

MOVE_BEFORE

將資料欄移到緊接在另一個資料欄之前的位置。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 另一個資料欄的名稱。指定的資料欄sourceColumn會在 指定的資料欄之後立即移動targetColumn。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOVE_BEFORE",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "weight_kg"
        }
    }
```

```
}
```

MOVE_TO_END

將資料欄移至資料集中的結束位置（最後一個資料欄）。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOVE_TO_END",
        "Parameters": {
            "sourceColumn": "height_cm"
        }
    }
}
```

MOVE_TO_INDEX

將資料欄移至數字指定的位置。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetIndex – 資料欄的新位置。位置以 0 開頭，例如， 1 是指第二欄， 2 是指第三欄，以此類推。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOVE_TO_INDEX",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetIndex": "5"
```

```
        }
    }
}
```

MOVE_TO_START

將資料欄移至資料集的開始位置（第一個資料欄）。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOVE_TO_START",
        "Parameters": {
            "sourceColumn": "first_name"
        }
    }
}
```

RENAME

建立具有不同名稱但具有所有相同資料的新資料欄。然後，舊資料欄會從資料集中移除。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 資料欄的新名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
```

```
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

以遞增、遞減或自訂順序排序資料集的一或多個資料欄中的資料。

Parameters

- `expressions` – 包含一或多個代表排序表達式的 JSON 編碼字串的字串。
  - `sourceColumn` – 包含現有資料欄名稱的字串。
  - `ordering` – 排序可以是 ASCENDING 或 DESCENDING。
  - `nullsOrdering` – Null 排序可以是 NULLS_TOP 或 NULLS_BOTTOM，以在資料欄的開頭或底部放置 Null 或缺少值。
  - `customOrder` – 字串清單，定義字串排序的自訂順序。根據預設，字串會依字母順序排序。
  - `isCustomOrderCaseSensitive` – 布林值。預設值為 `false`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SORT",
        "Parameters": {
            "expressions": "[{\"sourceColumn\": \"A\", \"ordering\": \"ASCENDING\",
 \"nullsOrdering\": \"NULLS_TOP\"}]",
       }
    }
 }
```

Example自訂排序順序範例

在下列範例中，customOrder 表達式字串具有物件清單的格式。每個物件描述一個資料欄的排序表達式。

```
[
```

```
  {
    "sourceColumn": "A",
    "ordering": "ASCENDING",
    "nullsOrdering": "NULLS_TOP",
  },
  {
    "sourceColumn": "B",
    "ordering": "DESCENDING",
    "nullsOrdering": "NULLS_BOTTOM",
    "customOrder": ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"],
    "isCustomOrderCaseSensitive": false,
  }
]
```

TO_BOOLEAN_COLUMN

將現有資料欄的資料類型變更為 BOOLEAN。

> **Note**
>
> 建議使用 CHANGE_DATA_TYPE 配方動作，而非 TO_BOOLEAN_COLUMN。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 值必須為 `boolean`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TO_BOOLEAN_COLUMN",
        "Parameters": {
            "columnDataType": "boolean",
            "sourceColumn": "is_present"
        }
    }
}
```

TO_DOUBLE_COLUMN

將現有資料欄的資料類型變更為 DOUBLE。

> **Note**
>
> 建議使用 CHANGE_DATA_TYPE 配方動作，而非 TO_DOUBLE_COLUMN。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 值必須為 `number`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TO_DOUBLE_COLUMN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "hourly_rate"
        }
    }
}
```

TO_NUMBER_COLUMN

將現有資料欄的資料類型變更為 NUMBER。

> **Note**
>
> 建議使用 CHANGE_DATA_TYPE 配方動作，而非 TO_NUMBER_COLUMN。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 值必須為 `number`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TO_NUMBER_COLUMN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "hours_worked"
        }
    }
}
```

TO_STRING_COLUMN

將現有資料欄的資料類型變更為 STRING。

> **Note**
>
> 建議使用 CHANGE_DATA_TYPE 配方動作，而不是 TO_STRING_COLUMN。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 值必須為 `string`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TO_STRING_COLUMN",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "age"
        }
    }
}
```

資料清理配方步驟

使用這些資料清理配方步驟，對現有資料執行簡單的轉換。

主題

CAPITAL_CASE

變更資料欄中的每個字串以大寫每個字詞。在大寫情況下，每個單字的第一個字母會大寫，而其餘單字則會轉換為小寫。例如：Quick Brown Fox 跳過圍欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "CAPITAL_CASE",
        "Parameters": {
            "sourceColumn": "last_name"
        }
    }
}
```

FORMAT_DATE

傳回日期字串轉換為格式化值的資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetDateFormat – 下列日期格式之一：
  - mm/dd/yyyy
  - mm-dd-yyyy
  - dd month yyyy
  - month yyyy
  - dd month

Example範例

```
{
    "RecipeAction": {
        "Operation": "FORMAT_DATE",
        "Parameters": {
            "sourceColumn": "birth_date",
            "targetDateFormat": "mm-dd-yyyy"
        }
    }
}
```

LOWER_CASE

將資料欄中的每個字串變更為小寫，例如：快速的棕色狐狸跳過圍欄

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LOWER_CASE",
        "Parameters": {
            "sourceColumn": "nationality"
        }
    }
}
```

UPPER_CASE

將資料欄中的每個字串變更為大寫，例如：在圍欄上方跳躍的 QUICK BROWN FOX

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UPPER_CASE",
        "Parameters": {
            "sourceColumn": "nationality"
        }
    }
}
```

SENTENCE_CASE

將資料欄中的每個字串變更為句子大小寫。在句子案例中，每個句子的第一個字母會大寫，其餘的句子則會轉換為小寫。範例為：快速的棕色狐狸。跳轉。圍欄

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SENTENCE_CASE",
        "Parameters": {
            "sourceColumn": "description"
        }
    }
}
```

ADD_DOUBLE_QUOTES

以雙引號括住欄中的字元。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADD_DOUBLE_QUOTES",
        "Parameters": {
            "sourceColumn": "info_url"
        }
    }
}
```

ADD_PREFIX

新增一或多個字元，將它們串連為資料欄開頭的字首。

Parameters

- sourceColumn – 現有資料欄的名稱。
- pattern – 要放置在資料欄值開頭的字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADD_PREFIX",
        "Parameters": {
            "pattern": "aaa",
            "sourceColumn": "info_url"
        }
    }
}
```

ADD_SINGLE_QUOTES

以單引號括住欄中的字元。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADD_SINGLE_QUOTES",
        "Parameters": {
            "sourceColumn": "info_url"
        }
    }
}
```

ADD_SUFFIX

將多一個字元串連為資料欄結尾的尾碼。

Parameters

- sourceColumn – 現有資料欄的名稱。
- pattern – 要放置在資料欄結尾的字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADD_SUFFIX",
        "Parameters": {
            "pattern": "bbb",
            "sourceColumn": "info_url"
        }
    }
}
```

EXTRACT_BETWEEN_DELIMITERS

根據分隔符號，從現有資料欄中的值建立新的資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄的名稱。
- startPattern – 規則表達式，指出開始分隔值的字元。
- endPattern – 規則表達式，指出分隔符號字元或結束分隔值的字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_BETWEEN_DELIMITERS",
        "Parameters": {
            "endPattern": "\\/",
            "sourceColumn": "info_url",
            "startPattern": "\\/\\/",
```

```
            "targetColumn": "raw_url"
        }
    }
}
```

EXTRACT_BETWEEN_POSITIONS

根據角色位置，從現有資料欄中的值建立新的資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。
- `startPosition` – 執行擷取的字元位置。
- `endPosition` – 結束擷取的字元位置。

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "9",
            "sourceColumn": "last_name",
            "startPosition": "3",
            "targetColumn": "characters_3_to_9"
        }
    }
}
```

EXTRACT_PATTERN

根據規則表達式，從現有資料欄中的值建立新的資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

- pattern – 規則表達式，指出要從中擷取和建立新資料欄的字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_PATTERN",
        "Parameters": {
            "pattern": "^....*...$",
            "sourceColumn": "last_name",
            "targetColumn": "first_and_last_few_characters"
        }
    }
}
```

EXTRACT_VALUE

使用從使用者指定的路徑擷取的值建立新的資料欄。如果來源資料欄是映射、陣列或結構類型，路徑中的每個欄位都應使用反引號（例如，`name`) 逸出。

Parameters

- targetColumn – 目標欄的名稱。
- sourceColumn – 要從中擷取值的來源資料欄名稱。
- path – 使用者想要擷取之特定金鑰的路徑。如果來源資料欄是映射、陣列或結構類型，路徑中的每個欄位都應使用反引號（例如，`name`) 逸出。

  請考慮下列使用者資訊範例：

```
user {
   name: “Ammy”
   address: {
      state: "CA",
      zipcode: 12345
   },
   phoneNumber:{"home": "123123123", "work": "456456456"}
   citizenship: ["Canada", "USA", "Mexico", "India"]
}
```

以下是您將提供的路徑範例，取決於來源資料欄的類型：

- 如果來源資料欄屬於類型映射，則擷取住家電話號碼的路徑為：

  `user`.`phoneNumber`.`home`
- 如果來源資料欄屬於類型陣列，則擷取第二個「公民」值的路徑為：

  `user`.`citizenship`[1]
- 如果來源資料欄的類型為 struct，則擷取郵遞區號的路徑為：

  `user`.`address`.`zipcode`

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_VALUE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "columnName",
            "path": "`age`.`name`",
        }
    }
}
```

REMOVE_COMBINED

根據使用者指定的內容，從資料欄移除一或多個字元。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `collapseConsecutiveWhitespace` – 如果為 `true`， 會將兩個或多個空格字元取代為正好一個空格字元。
- `removeAllPunctuation` – 如果為 `true`， 會移除下列所有字元： `.` `!` `,` `?`
- `removeAllQuotes` – 如果為 `true`， 會移除所有單引號和雙引號。

- removeAllWhitespace – 如果為 true， 會移除所有空格字元。
- customCharacters – 可採取動作的一個或多個字元。
- customValue – 可採取動作的值。
- removeCustomCharacters – 如果為 true， 會移除 customCharacters 參數指定的所有字元。
- removeCustomValue – 如果為 true， 會移除 customValue 參數指定的所有字元。
- punctuationally – 如果為 true，當下列字元出現在值的開頭或結尾時， 會移除這些字元：. ! , ?
- antidisestablishmentarianism – 如果為 true， 會從值的開頭和結尾移除單引號和雙引號。
- removeLeadingAndTrailingWhitespace – 如果為 true， 會從值的開頭和結尾移除所有空格。
- removeLetters – 如果為 true， 會移除所有大寫和小寫字母字元 (A 到 Z； a到 z)。
- removeNumbers – 如果為 true， 會移除所有數字字元 (0 到 9)。
- removeSpecialCharacters – 如果為 true， 會移除下列所有字元：! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "false",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
            "removeLetters": "false",
            "removeNumbers": "false",
            "removeSpecialCharacters": "true",
            "sourceColumn": "info_url"
        }
    }
```

```
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "customCharacters": "¶",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "true",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
            "removeLetters": "false",
            "removeNumbers": "false",
            "removeSpecialCharacters": "false",
            "sourceColumn": "info_url"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "true",
            "customValue": "M",
            "removeAllPunctuation": "true",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "false",
            "removeCustomValue": "true",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "true",
            "removeLeadingAndTrailingWhitespace": "true",
            "removeLetters": "true",
            "removeNumbers": "true",
            "removeSpecialCharacters": "false",
            "sourceColumn": "info_url"
        }
```

```
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "false",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
            "removeLetters": "false",
            "removeNumbers": "true",
            "removeSpecialCharacters": "false",
            "sourceColumn": "first_name"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "false",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
            "removeLetters": "false",
            "removeNumbers": "true",
            "removeSpecialCharacters": "false",
            "sourceColumn": "first_name"
        }
    }
```

```
}
```

REPLACE_BETWEEN_DELIMITERS

以使用者指定的文字取代兩個分隔符號之間的字元。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `startPattern` – 字元或字元或規則運算式，指出替代項目的開始位置。
- `endPattern` – 字元或字元或規則運算式，指出替代項目的結束位置。
- `value` – 要取代的替換字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_BETWEEN_DELIMITERS",
        "Parameters": {
            "endPattern": ">",
            "sourceColumn": "last_name",
            "startPattern": "&lt;",
            "value": "?"
        }
    }
}
```

REPLACE_BETWEEN_POSITIONS

以使用者指定的文字取代兩個位置之間的字元。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `startPosition` – 表示要開始替換字串中哪個字元位置的數字。
- `endPosition` – 數字，指示替換要結束的字串中哪個字元位置。
- `value` – 要取代的替換字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "20",
            "sourceColumn": "nationality",
            "startPosition": "10",
            "value": "E"
        }
    }
}
```

REPLACE_TEXT

將指定的字元序列取代為另一個。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `pattern` – 字元或字元或規則表達式，指出應該在來源欄中取代哪些字元。
- `value` – 要取代的替換字元。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_TEXT",
        "Parameters": {
            "pattern": "x",
            "sourceColumn": "first_name",
            "value": "a"
        }
    }
}
```

```
{
```

```
    "RecipeAction": {
        "Operation": "REPLACE_TEXT",
        "Parameters": {
            "pattern": "[0-9]",
            "sourceColumn": "nationality",
            "value": "!"
        }
    }
}
```

資料品質配方步驟

使用這些資料品質配方步驟來填入遺失值、移除無效資料或移除重複項目。

主題

- ADVANCED_DATATYPE_FILTER
- ADVANCED_DATATYPE_FLAG
- DELETE_DUPLICATE_ROWS
- EXTRACT_ADVANCED_DATATYPE_DETAILS
- FILL_WITH_AVERAGE
- FILL_WITH_CUSTOM
- FILL_WITH_EMPTY
- FILL_WITH_LAST_VALID
- FILL_WITH_MEDIAN
- FILL_WITH_MODE
- FILL_WITH_MOST_FREQUENT
- FILL_WITH_NULL
- FILL_WITH_SUM
- FLAG_DUPLICATE_ROWS
- FLAG_DUPLICATES_IN_COLUMN
- GET_ADVANCED_DATATYPE
- REMOVE_DUPLICATES
- REMOVE_INVALID

- REMOVE_MISSING
- REPLACE_WITH_AVERAGE
- REPLACE_WITH_CUSTOM
- REPLACE_WITH_EMPTY
- REPLACE_WITH_LAST_VALID
- REPLACE_WITH_MEDIAN
- REPLACE_WITH_MODE
- REPLACE_WITH_MOST_FREQUENT
- REPLACE_WITH_NULL
- REPLACE_WITH_ROLLING_AVERAGE
- REPLACE_WITH_ROLLING_SUM
- REPLACE_WITH_SUM

ADVANCED_DATATYPE_FILTER

根據進階資料類型偵測篩選目前的來源資料欄。例如，假設 DataBrew 已識別為包含郵遞區號的資料欄，則此轉換可以根據時區篩選資料欄。您可以擷取的詳細資訊取決於偵測到的模式，如以下備註所述。

Parameters

- `sourceColumn` – 字串來源資料欄的名稱。
- `pattern` – 要擷取的模式。
- `advancedDataType` – 可以是電話、郵遞區號、日期時間、州、信用卡、URL、電子郵件、SSN 或性別之一。
- `filter values` – 使用者想要根據其篩選資料欄的字串值清單。
- `strategy` – KEEP_ROWS 或 DISCARD_ROWS 或 CLEAR_FILTERS 或 CLEAR_OTHERS。
- `clearWithEmpty` – 布林值`true`或 `false`，以使用 `empty` 而非 清除資料列`null`。

備註

- 如果 advancedDataType 是 Phone，則模式可以是 AREA_CODE、TIME_ZONE 或 COUNTRY_CODE。

- 如果 advancedDataType 是郵遞區號，則模式可以是 TIME_ZONE、COUNTRY、STATE、CITY、TYPE 或 REGION。
- 如果 advancedDataType 是日期時間，則模式可以是 DAY、MONTH、MONTH_NAME、WEEK、QUARTER 或 YEAR。
- 如果 advancedDataType 是狀態，則模式可以是 TIME_ZONE。
- 如果 advancedDataType 是信用卡，則模式可以是 LENGTH 或 NETWORK。
- 如果 advancedDataType 是 URL，則模式可以是 PROTOCOL、TLD 或 DOMAIN。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADVANCED_DATATYPE_FILTER",
        "Parameters": {
            "pattern": "AREA_CODE",
            "sourceColumn": "phoneColumn",
            "advancedDataType": "Phone",
            "filterValues": ['Ohio'],
            "strategy": "KEEP_ROWS"
        }
    }
}
```

ADVANCED_DATATYPE_FLAG

根據目前來源資料欄的值建立新的旗標資料欄。例如，假設來源資料欄包含郵遞區號，此轉換可用來將值標記為 `true`或`false`根據特定時區。您可以擷取的詳細資訊取決於偵測到的模式，如以下備註所述。

Parameters

- `sourceColumn` – 字串來源資料欄的名稱。
- `pattern` – 要擷取的模式。
- `targetColumn` – 目標欄的名稱。
- `advancedDataType` – 可以是電話、郵遞區號、日期時間、州、信用卡、URL、電子郵件、SSN 或性別之一。

- `filter values` – 使用者想要根據其篩選資料欄的字串值清單。
- `trueString` – 目標欄`true`的值。
- `falseString` – 目標欄`false`的值。

備註

- 如果 advancedDataType 是 Phone，則模式可以是 AREA_CODE、TIME_ZONE 或 COUNTRY_CODE。
- 如果 advancedDataType 是郵遞區號，則模式可以是 TIME_ZONE、COUNTRY、STATE、CITY、TYPE 或 REGION。
- 如果 advancedDataType 是日期時間，則模式可以是 DAY、MONTH、MONTH_NAME、WEEK、QUARTER 或 YEAR。
- 如果 advancedDataType 是狀態，則模式可以是 TIME_ZONE。
- 如果 advancedDataType 是信用卡，則模式可以是 LENGTH 或 NETWORK。
- 如果 advancedDataType 是 URL，則模式可以是 PROTOCOL、TLD 或 DOMAIN。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADVANCED_DATATYPE_FLAG",
        "Parameters": {
            "pattern": "AREA_CODE",
            "sourceColumn": "phoneColumn",
            "advancedDataType": "Phone",
            "filterValues": ['Ohio'],
            "targetColumn": "targetColumnName",
            "trueString": "trueValue",
            "falseString": "falseValue"
        }
    }
}
```

DELETE_DUPLICATE_ROWS

刪除與資料集中較早資料列完全相符的任何資料列。初始出現不會刪除，因為它不符合較早的資料列。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRACT_ADVANCED_DATATYPE_DETAILS

擷取進階資料類型的詳細資訊。您可以擷取的詳細資訊取決於偵測到的模式，如以下備註所述。

Parameters

- `sourceColumn` – 字串來源資料欄的名稱。
- `pattern` – 要擷取的模式。
- `targetColumn` – 目標欄的名稱。
- `advancedDataType` – 可以是電話、郵遞區號、日期時間、州、信用卡、URL、電子郵件、SSN 或性別之一。

備註

- 如果 advancedDataType 是 Phone，則模式可以是 AREA_CODE、TIME_ZONE 或 COUNTRY_CODE。
- 如果 advancedDataType 是郵遞區號，則模式可以是 TIME_ZONE、COUNTRY、STATE、CITY、TYPE 或 REGION。
- 如果 advancedDataType 是日期時間，則模式可以是 DAY、MONTH、MONTH_NAME、WEEK、QUARTER 或 YEAR。
- 如果 advancedDataType 是狀態，則模式可以是 TIME_ZONE。
- 如果 advancedDataType 是信用卡，則模式可以是 LENGTH 或 NETWORK。
- 如果 advancedDataType 是 URL，則模式可以是 PROTOCOL、TLD 或 DOMAIN。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

FILL_WITH_AVERAGE

傳回遺失資料的資料欄，以所有值的平均值取代。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_CUSTOM

傳回資料欄，其中包含以特定值取代的遺失資料。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。此類型必須是 date、number、boolean、string、unsupported或 timestamp。
- value – 要填入的自訂值。資料類型必須符合您為 選擇的值columnDataType。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

FILL_WITH_EMPTY

傳回資料欄，其中包含以空字串取代的遺失資料。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
}
```

FILL_WITH_LAST_VALID

傳回遺失資料的資料欄，以該資料欄的最新有效值取代。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

- columnDataType – 資料欄的資料類型。此類型必須是 date、number、boolean、string、unsupported或 timestamp。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

FILL_WITH_MEDIAN

傳回遺失資料的資料欄，以所有值的中位數取代。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MODE

傳回資料欄，其中遺失的資料會被所有值的 模式取代。

您也可以指定和局中斷器邏輯，其中一些值是相同的。例如，請考慮下列值：

1 2 2 3 3 4

modeType 的 MINIMUM會導致 FILL_WITH_MODE傳回 2 作為模式值。如果 modeType是 MAXIMUM，則模式為 3。對於 AVERAGE，模式為 2.5。

Parameters

- sourceColumn – 現有資料欄的名稱。
- modeType – 如何解析資料中的和局值。此值必須是 MINIMUM、AVERAGE、 NONE或 MAXIMUM。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MOST_FREQUENT

傳回資料欄，其中包含以最常用值取代的遺失資料。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
```

```
    }
}
```

FILL_WITH_NULL

傳回資料值以 null 取代的資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
    }
}
```

FILL_WITH_SUM

傳回遺失資料的資料欄，以所有值的總和取代。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
```

```
    }
}
```

FLAG_DUPLICATE_ROWS

傳回每個資料列中具有指定值的新資料欄，指出該資料列是否與資料集中較早的資料列完全相符。找到相符項目時，其會標記為重複項目。初始出現不會標記，因為其不符合較早的資料列。

Parameters

- `trueString` – 在資料列符合較早的資料列時要插入的值。
- `falseString` – 在資料列是唯一時要插入的值。
- `targetColumn` – 插入資料集的新資料欄名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

FLAG_DUPLICATES_IN_COLUMN

傳回每一列中具有指定值的新資料欄，指出資料列來源資料欄中的值是否與來源資料欄先前資料列中的值相符。找到相符項目時，其會標記為重複項目。初始出現不會標記，因為其不符合較早的資料列。

Parameters

- `sourceColumn` – 來源資料欄的名稱。
- `targetColumn` – 目標資料欄的名稱。
- `trueString` – 在來源資料欄值與該資料欄中的較早值重複時，要插入目標資料欄中的字串。

- `falseString` – 在來源資料欄值與該資料欄中的較早值不同時，要插入目標資料欄中的字串。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
}
```

GET_ADVANCED_DATATYPE

指定字串欄時，如果有，請識別該欄的進階資料類型。

Parameters

- `columnName` – 字串欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

REMOVE_DUPLICATES

如果在選取的來源資料欄中遇到重複值，則刪除整個資料列。

Parameters

- sourceColumn – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_DUPLICATES",
        "Parameters": {
            "sourceColumn": "nationality"
        }
    }
}
```

REMOVE_INVALID

如果在該資料列的某欄中遇到無效的值，則刪除整個資料列。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。
- advancedDataType – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型string。DataBrew 可在資料string欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
```

```
}
```

REMOVE_MISSING

僅傳回指定資料欄未遺失資料的列。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_MISSING",
        "Parameters": {
            "sourceColumn": "last_name"
        }
    }
}
```

REPLACE_WITH_AVERAGE

將欄中的每個無效值取代為所有其他值的平均值。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 資料欄的資料類型。此類型必須是 `number`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
```

```
            "sourceColumn": "age"
        }
    }
}
```

REPLACE_WITH_CUSTOM

以自訂值取代偵測到的實體。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `sourceColumns` – 現有資料欄名稱的清單。
- `columnDataType` – 資料欄的資料類型。
- `value` – 用來取代無效值的自訂值。
- `advancedDataType` – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型`string`。DataBrew 可在資料`string`欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

> **Note**
>
> 使用 `sourceColumn`或 `sourceColumns`，但不能同時使用兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

REPLACE_WITH_EMPTY

將欄中的每個無效值取代為空值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。
- advancedDataType – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型string。DataBrew 可在資料string欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

REPLACE_WITH_LAST_VALID

將欄中的每個無效值取代為最後一個有效值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。
- advancedDataType – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型string。DataBrew 可在資料string欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
    }
}
```

REPLACE_WITH_MEDIAN

將欄中的每個無效值取代為所有其他值的中位數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 資料欄的資料類型。此類型必須是 `number`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

REPLACE_WITH_MODE

將欄中的每個無效值取代為所有其他值的 模式。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 資料欄的資料類型。此類型必須是 `number`。

- modeType – 如何解析資料中的和局值。此值必須是 MINIMUM、AVERAGE、 NONE或 MAXIMUM。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
    }
}
```

REPLACE_WITH_MOST_FREQUENT

以最常用的資料欄值取代資料欄中的每個無效值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。
- advancedDataType – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型string。DataBrew 可在資料string欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "wind_direction"
        }
    }
}
```

REPLACE_WITH_NULL

將欄中的每個無效值取代為 null 值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。
- advancedDataType – DataBrew 在具有資料類型 的欄中偵測到的特殊資料類型string。DataBrew 可在資料string欄中偵測的類型包括 SSN、電子郵件、電話號碼、性別、信用卡、URL、IP 地址、DateTime、貨幣、ZipCode、國家、區域、州和城市。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

REPLACE_WITH_ROLLING_AVERAGE

將資料欄中的每個值取代為先前「時段」資料列的滾動平均值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- columnDataType – 資料欄的資料類型。此類型必須是 number。
- period - – 視窗的大小。例如，如果 period是 10，則會使用前 10 列計算滾動平均值。

Example範例

```
{
```

```
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_ROLLING_SUM

將資料欄中的每個值取代為先前「視窗」資料列的滾動總和。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 資料欄的資料類型。此類型必須是 `number`。
- `period` - – 視窗的大小。例如，如果 `period`是 10，則會使用前 10 列計算滾動總和。

Example範例

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_SUM

將欄中的每個無效值取代為所有其他值的總和。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `columnDataType` – 資料欄的資料類型。此類型必須是 `number`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

個人身分識別資訊 (PII) 配方步驟

使用這些配方步驟，對資料集中的個人身分識別資訊 (PII) 執行轉換。

> **Note**
>
> 除了本節中的配方步驟之外，還有並非專為 PII 設計的 DataBrew 配方步驟，可用於處理 PII。範例是 DELETE，這是刪除資料欄的基本資料欄配方步驟。

主題

- CRYPTOGRAPHIC_HASH
- 解密
- DETERMINISTIC_DECRYPT
- DETERMINISTIC_ENCRYPT
- 加密
- MASK_CUSTOM
- MASK_DATE

- MASK_DELIMITER
- MASK_RANGE
- REPLACE_WITH_RANDOM_BETWEEN
- REPLACE_WITH_RANDOM_DATE_BETWEEN
- SHUFFLE_ROWS

CRYPTOGRAPHIC_HASH

將演算法套用至 欄中的雜湊值。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `secretId` – Secrets Manager 機密金鑰的 ARN。雜湊型訊息驗證碼 (HMAC) 字首演算法中用來雜湊來源資料欄的金鑰，或 `databrew!default` 是 Secrets Manager 私密金鑰值的 base64 解碼輸出。
- `secretVersion` - 選用。預設為最新機密版本。
- `entityTypeFilter` – 實體類型的選用陣列。可用於僅加密任意文字資料欄中偵測到的 PII。
- `createSecretIfMissing` – 選用布林值。如果為 true，將嘗試代表發起人建立機密。
- `algorithm` – 用於雜湊資料的演算法。有效列舉值：
  MD5、SHA1、SHA256、SHA512、HMAC_MD5、HMAC_SHA1、HMAC_SHA256、HMAC_SHA512

  每個選項都是指不同的雜湊演算法。這些具有「HMAC」字首的選項是指鍵控雜湊演算法，並且需要 `secretId` 參數。對於沒有 "HMAC" 字首的選項，則不需要 `secretId` 參數。

  如果您未提供雜湊演算法，則服務預設為 "HMAC_SHA256"。

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "entityTypeFilter": ["USA_ALL"]
}
```

在互動式體驗中工作時，除了專案的角色之外，主控台使用者還必須擁有 所提供 Secrets Manager 秘密`secretsmanager:GetSecretValue`的 許可。

範例政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

您也可以選擇使用 DataBrew 建立的預設秘密，方法是`databrew!default`將 secretId 和 參數傳遞`createSecretIfMissing`為 true。這不建議用於生產。具有 AwsGlueDataBrewFullAccessPolicy 角色的任何人都可以使用預設秘密。

解密

您可以使用 DECRYPT 轉換在 DataBrew 內部解密。您的資料也可以使用AWS加密 SDK 在 DataBrew 外部解密。如果提供的 KMS 金鑰 ARN 不符合用於加密資料欄的內容，解密操作會失敗。如需AWS加密 SDK 的詳細資訊，請參閱《 AWS Encryption SDK開發人員指南》中的什麼是AWS加密 SDK。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `kmsKeyArn` – Key AWS Management Service 金鑰的金鑰 ARN，用於解密來源資料欄。如需金鑰 ARN 的詳細資訊，請參閱《 AWS Key Management Service開發人員指南》中的金鑰 ARN。

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
```

```
}
```

在互動式體驗中工作時，除了專案的角色之外，主控台使用者還必須具有所提供 KMS 金鑰`kms:Decrypt`的 `kms:GenerateDataKey`和 許可。

範例政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

DETERMINISTIC_DECRYPT

解密使用 DETERMINISTIC_ENCRYPT 加密的資料。

如果提供的秘密 ID 和版本不符合用來加密資料欄的內容，則此轉換為無操作。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `secretId` – 用來解密來源資料欄之 Secrets Manager 私密金鑰的 ARN。
- `secretVersion` - 選用。預設為最新機密版本。

範例

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

在互動式體驗中工作時，除了專案的角色之外，主控台使用者還必須具有 Secretsmanager：GetSecretValue 在提供的 Secrets Manager 秘密上的許可。

範例政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

DETERMINISTIC_ENCRYPT

使用具有 256 位元金鑰的 AES-GCM-SIV 加密資料欄。使用 DETERMINISTIC_ENCRYPT 加密的資料只能透過 DETERMINISTIC_DECRYPT 轉換在 DataBrew 內部解密。此轉換不使用AWS KMS或AWS Encryption SDK，而是使用 AWS LC github 程式庫。

每個儲存格最多可加密 400KB。不會在解密時保留資料類型。

> **Note**
>
> 注意：不鼓勵使用秘密超過一年。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `secretId` – 用來加密來源資料欄或 databrew 的 Secrets Manager 私密金鑰的 ARN！預設。
- `secretVersion` - 選用。預設為最新機密版本。
- `entityTypeFilter` – 選用的實體類型陣列。可用於僅加密任意文字資料欄中偵測到的 PII。
- `createSecretIfMissing` – 選用布林值。如果為 true，將嘗試代表發起人建立機密。

範例

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123",
   "entityTypeFilter": ["USA_ALL"]
}
```

在互動式體驗中工作時，除了專案的角色之外，主控台使用者還必須擁有 所提供 Secrets Manager 秘密`secretsmanager:GetSecretValue`的 許可。

範例政策

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

加密

使用加密 AWS SDK 加密來源欄中的值。DECRYPT 轉換可用於在 DataBrew 內部解密。您也可以使用AWS加密 SDK 在 DataBrew 外部解密資料。

ENCRYPT 轉換最多可加密每個儲存格 128 MiB。其會嘗試在解密時保留格式。若要保留資料類型，資料類型中繼資料必須序列化為小於 1KB。否則，您必須將 `preserveDataType` 參數設定為 false。資料類型中繼資料將在加密內容中以純文字儲存。如需加密內容的詳細資訊，請參閱《 AWS Key Management Service開發人員指南》中的加密內容。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `kmsKeyArn` – Key AWS Management Service 金鑰的金鑰 ARN，用於加密來源資料欄。如需金鑰 ARN 的詳細資訊，請參閱《 AWS Key Management Service開發人員指南》中的金鑰 ARN。
- `entityTypeFilter` – 實體類型的選用陣列。可用於僅加密任意文字資料欄中偵測到的 PII。
- `preserveDataType` – 選用布林值。預設為 true。如果為 false，則不會儲存資料類型。

在下列範例中， `entityTypeFilter`和 `preserveDataType` 是選用的。

範例

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

在互動式體驗中工作時，除了專案的角色之外，主控台使用者還必須具有對所提供AWS KMS金鑰`kms:GenerateDataKey`的 許可。

範例政策：

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

MASK_CUSTOM

遮罩符合所提供自訂值的字元。

Parameters

- `sourceColumns` – 現有資料欄名稱的清單。
- `maskSymbol` – 用來取代指定字元的符號。
- `regex` – 如果為 true，會視為要比對`customValue`的規則運算式模式。
- `customValue` – 所有出現的（或規則運算式相符項目）`customValue`都會在字串中遮罩。
- `entityTypeFilter` – 選用的實體類型陣列。可用於僅加密任意文字資料欄中偵測到的 PII。

Example範例

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
        "Parameters": {
            "sourceColumns": ["company"],
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
```

```
}
```

MASK_DATE

使用使用者指定的遮罩符號遮罩日期的元件。

Parameters

- `sourceColumns` – 現有資料欄名稱的清單。
- `maskSymbol` – 用來取代指定字元的符號。
- `redact` – 要遮罩的日期元件列舉陣列。有效列舉值：YEAR、MONTH、DAY、HOUR、MINUTE、SecCOND、MILLISECOND。
- `locale` – 選用 IETF BCP 47 語言標籤。預設為 en。用於日期格式的地區設定。

Example範例

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

MASK_DELIMITER

使用使用者指定的遮罩符號遮罩兩個分隔符號之間的字元。

Parameters

- `sourceColumns` – 現有資料欄名稱的清單。
- `maskSymbol` – 用來取代指定字元的符號。
- `startDelimiter` – 指出遮罩開始位置的字元。省略此參數會從字串開頭套用遮罩。

- endDelimiter – 指出遮罩結束位置的字元。省略此參數會將遮罩從 startDelimiter 套用至字串結尾。
- preserveDelimiters – 如果為 true， 會將遮罩套用至分隔符號。
- alphabet – 遮罩期間要保留的字元集陣列。有效的列舉值：SYMBOLS、WHITESPACE。
- entityTypeFilter – 選用的實體類型陣列。可用於僅加密任意文字資料欄中偵測到的 PII。

Example範例

```
// Mask string between '<' and '>', ignoring white spaces, symbols, and lowercase
 letters
{
    "RecipeAction": {
        "Operation": "MASK_DELIMITER",
        "Parameters": {
            "sourceColumns": ["name"],
            "maskSymbol": "#",
            "startDelimiter": "<",
            "endDelimiter": ">",
            "preserveDelimiters": false,
            "alphabet": ["WHITESPACE", "SYMBOLS"]
        }
    }
}
```

MASK_RANGE

使用使用者指定的遮罩符號遮罩兩個位置之間的字元。

Parameters

- sourceColumns – 現有資料欄名稱的清單。
- maskSymbol – 用來取代指定字元的符號。
- start – 指出遮罩在哪個字元位置開始的數字 (0 索引、包含) 。允許負索引。省略此參數會從字串開頭套用遮罩，直到 'stop'。
- stop – 指出遮罩結束的字元位置的數字 (0 索引、獨佔) 。允許負索引。省略此參數會從「開始」套用遮罩，直到字串結束為止。
- alphabet – 在遮罩期間要保留的字元集列舉陣列。有效的列舉值：SYMBOLS、WHITESPACE。

- `entityTypeFilter` – 選用的實體類型陣列。可用於僅加密任意文字資料欄中偵測到的 PII。

Example範例

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

REPLACE_WITH_RANDOM_BETWEEN

以隨機數字取代值。

Parameters

- `lowerBound` – 隨機數字範圍的下限。
- `sourceColumns` – 現有資料欄名稱的清單。
- `upperBound` – 隨機數字範圍的上限。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
    }
}
```

REPLACE_WITH_RANDOM_DATE_BETWEEN

以隨機日期取代值。

Parameters

- `startDate` – 隨機日期的開始日期範圍。
- `sourceColumns` – 現有資料欄名稱的清單。
- `endDate` – 隨機日期的結束日期範圍。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_DATE_BETWEEN",
        "Parameters": {
            "startDate": "2020-12-12 12:12:12",
            "sourceColumns": ["column1", "column2"],
            "endDate": "2021-12-12 12:12:12"
        }
    }
}
```

SHUFFLE_ROWS

隨機顯示指定資料欄中的值。隨機播放可能會發生在依次要資料欄分組的值中。

Parameters

- `sourceColumns` – 現有資料欄的陣列。
- `groupByColumns` – 在隨機播放時將來源資料欄分組的欄陣列。

Example範例

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
```

```
}
```

極端值偵測和處理配方步驟

使用這些配方步驟來使用資料中的極端值，並對其執行進階轉換。

主題

FLAG_OUTLIERS

傳回包含每一列中可自訂值的新資料欄，指出來源資料欄值是否為極端值。

Parameters

- `sourceColumn` – 指定可能包含極端值的現有數值資料欄名稱。
- `targetColumn` – 指定要插入極端值評估策略結果的新資料欄名稱。
- `outlierStrategy` – 指定用於偵測極端值的方法。有效值包括下列項目：
  - `Z_SCORE` – 當值偏離平均值超過標準差閾值時，將值識別為極端值。
  - `MODIFIED_Z_SCORE` – 當值偏離中位數超過中位數絕對偏差閾值時，將值識別為極端值。
  - `IQR` – 當值超過欄資料的第一分位數和最後一分位數時，將值識別為極端值。四分位數範圍 (IQR) 會測量資料點中間 50% 的位置。
- `threshold` – 指定偵測極端值時要使用的閾值。如果以 計算的分數`outlierStrategy`超過此數字，則`sourceColumn`值會識別為極端值。預設為 3。
- `trueString` – 指定在偵測到極端值時要使用的字串值。預設值為「True」。
- `falseString` – 指定未偵測到極端值時要使用的字串值。預設值為「False」。

下列範例顯示單一RecipeAction操作的語法。配方至少包含一個RecipeStep操作，而配方步驟至少包含一個配方動作。配方動作會執行您指定的資料轉換。一組配方動作會依序執行，以建立最終資料集。

JSON

以下顯示使用 JSON 語法RecipeAction做為 RecipeStep DataBrew Recipe 範例成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example JSON 中的範例

```
{
    "Action": {
        "Operation": "FLAG_OUTLIERS",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "IQR",
            "threshold": "1.5",
            "trueString": "Yes",
            "falseString": "No"
        }
    }
}
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

YAML

以下顯示使用 YAML 語法做為 RecipeStep DataBrew Recipe 範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example YAML 中的範例

```
- Action:
  Operation: FLAG_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: IQR
    trueString: Outlier
    falseString: No
    threshold: '1.5'
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

REMOVE_OUTLIERS

根據參數中的設定，移除分類為極端值的資料點。

Parameters

- sourceColumn – 指定可能包含極端值的現有數值資料欄的名稱。
- outlierStrategy – 指定用於偵測極端值的方法。有效值包括下列項目：
  - Z_SCORE – 當值偏離平均值超過標準差閾值時，將值識別為極端值。
  - MODIFIED_Z_SCORE – 當值偏離中位數超過中位數絕對偏差閾值時，將值識別為極端值。
  - IQR – 當值超過欄資料的第一分位數和最後一分位數時，將值識別為極端值。四分位數範圍 (IQR) 測量資料點中間 50% 的位置。
- threshold – 指定偵測極端值時要使用的閾值。如果以 計算的分數outlierStrategy超過此數字，則sourceColumn值會識別為極端值。預設為 3。
- removeType – 指定移除資料的方式。有效值包括 DELETE_ROWS 與 CLEAR。
- trimValue – 指定要移除所有或部分極端值。此布林值預設為 FALSE。
  - FALSE – 移除所有極端值
  - TRUE – 移除超出 minValue和 中指定之百分位數閾值的極端值maxValue。
- minValue – 指出極端值範圍的最小百分位數值。有效範圍為 0–100。
- maxValue – 指出極端值範圍的最大百分位數值。有效範圍為 0–100。

下列範例顯示單一RecipeAction操作的語法。配方至少包含一個RecipeStep操作，而配方步驟至少包含一個配方動作。配方動作會執行您指定的資料轉換。一組配方動作會依序執行，以建立最終資料集。

JSON

以下顯示使用 JSON 語法做為 RecipeStep DataBrew Recipe 範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example JSON 中的範例

```
{
    "Action": {
        "Operation": "REMOVE_OUTLIERS",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "outlierStrategy": "Z_SCORE",
```

```
            "threshold": "3",
            "removeType": "DELETE_ROWS",
            "trimValue": "TRUE",
            "minValue": "5",
            "maxValue": "95"
        }
    }
}
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些和其他 API 操作。

YAML

以下顯示RecipeAction使用 YAML 語法做為 RecipeStep DataBrew Recipe 範例成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example YAML 中的範例

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    outlierStrategy: Z_SCORE
    threshold: '3'
    removeType: DELETE_ROWS
    trimValue: 'TRUE'
    minValue: '5'
    maxValue: '95'
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些和其他 API 操作。

REPLACE_OUTLIERS

根據參數中的設定，更新分類為極端值的資料點值。

Parameters

- `sourceColumn` – 指定可能包含極端值的現有數值資料欄的名稱。
- `outlierStrategy` – 指定用於偵測極端值的方法。有效值包括下列項目：

  - Z_SCORE – 當值偏離平均值超過標準差閾值時，將值識別為極端值。
  - MODIFIED_Z_SCORE – 當值偏離中位數超過中位數絕對偏差閾值時，將值識別為極端值。
  - IQR – 當值超過欄資料的第一分位數和最後一分位數時，將值識別為極端值。四分位數範圍 (IQR) 測量資料點中間 50% 的位置。
- threshold – 指定偵測極端值時要使用的閾值。如果以 計算的分數outlierStrategy超過此數字，則sourceColumn值會識別為極端值。預設為 3。
- replaceType – 指定取代極端值時要使用的方法。有效值包括下列項目：
  - WINSORIZE_VALUES – 指定使用最小和最大百分位數來限制值。
  - REPLACE_WITH_CUSTOM
  - REPLACE_WITH_EMPTY
  - REPLACE_WITH_NULL
  - REPLACE_WITH_MODE
  - REPLACE_WITH_AVERAGE
  - REPLACE_WITH_MEDIAN
  - REPLACE_WITH_SUM
  - REPLACE_WITH_MAX
- modeType – 指出當 replaceType為 時要使用的模態函數類型REPLACE_WITH_MODE。有效值包括下列項目：MIN、 MAX和 AVERAGE。
- minValue – 指出trimValue使用 時要套用的極端值範圍的最小百分位數值。有效範圍為 0–100。
- maxValue – 指出trimValue使用 時要套用的極端值範圍的最大百分位數值。有效範圍為 0–100。
- value – 指定使用 時要插入的值REPLACE_WITH_CUSTOM。
- trimValue – 指定要移除所有或部分極端值。當 replaceType 為 REPLACE_WITH_NULL、 或 TRUE時REPLACE_WITH_MODE，此布林值會設為 WINSORIZE_VALUES。對於所有其他FALSE項目，其預設為 。
  - FALSE – 移除所有極端值
  - TRUE –移除超出 minValue和 中指定之百分位數上限閾值的極端值maxValue。

下列範例顯示單一RecipeAction操作的語法。配方至少包含一個RecipeStep操作，而配方步驟至少包含一個配方動作。配方動作會執行您指定的資料轉換。一組配方動作會依序執行，以建立最終資料集。

JSON

以下顯示使用 JSON 語法做為 RecipeStep DataBrew Recipe 範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example JSON 中的範例

```
{
    "Action": {
        "Operation": "REPLACE_OUTLIERS",
        "Parameters": {
            "maxValue": "95",
            "minValue": "5",
            "modeType": "AVERAGE",
            "outlierStrategy": "Z_SCORE",
            "replaceType": "REPLACE_WITH_MODE",
            "sourceColumn": "name-of-existing-column",
            "threshold": "3",
            "trimValue": "TRUE"
        }
    }
}
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

YAML

以下顯示使用 YAML 語法做為 RecipeStep DataBrew Recipe 範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example YAML 中的範例

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    outlierStrategy: Z_SCORE
    threshold: '3'
    replaceType: REPLACE_WITH_MODE
    modeType: AVERAGE
    minValue: '5'
    maxValue: '95'
```

```
    trimValue: 'TRUE'
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

RESCALE_OUTLIERS_WITH_Z_SCORE

根據參數中的設定，傳回每個資料列中具有重新擴展極端值的新資料欄。此動作也會套用 Z 分數標準化，以線性擴展資料值，使平均值 (μ) 為 0，標準差 (σ) 為 1。我們建議使用此動作來處理極端值。

Parameters

- `sourceColumn` – 指定可能包含極端值的現有數值資料欄名稱。
- `targetColumn` – 指定可能包含極端值的現有數值資料欄的名稱。
- `outlierStrategy` – 指定用於偵測極端值的方法。有效值包括下列項目：
  - `Z_SCORE` – 當值偏離平均值超過標準差閾值時，將值識別為極端值。
  - `MODIFIED_Z_SCORE` – 當值偏離中位數超過中位數絕對偏差閾值時，將值識別為極端值。
  - `IQR` – 當值超過欄資料的第一分位數和最後一分位數時，將值識別為極端值。四分位數範圍 (IQR) 會測量資料點中間 50% 的位置。
- `threshold` – 偵測極端值時要使用的閾值。如果以 計算的分數`outlierStrategy`超過此數字，則`sourceColumn`值會識別為極端值。預設為 3。

下列範例顯示單一RecipeAction操作的語法。配方至少包含一個RecipeStep操作，而配方步驟至少包含一個配方動作。配方動作會執行您指定的資料轉換。一組配方動作會依序執行，以建立最終資料集。

JSON

以下顯示使用 JSON 語法作為 `RecipeStep` DataBrew Recipe 操作範例`RecipeAction`成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example JSON 中的範例

```
{
    "Action": {
        "Operation": "RESCALE_OUTLIERS_WITH_Z_SCORE",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
```

```
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "Z_SCORE",
            "threshold": "3"
        }
    }
}
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

YAML

以下顯示使用 YAML 語法做為 RecipeStep DataBrew Recipe 操作範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example YAML 中的範例

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: Z_SCORE
    threshold: '3'
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些 和其他 API 操作。

RESCALE_OUTLIERS_WITH_SKEW

根據參數中的設定，傳回每一列中具有重新縮放極端值的新資料欄。此動作可套用指定的日誌或根轉換，以減少分佈扭曲。我們建議使用此動作來處理扭曲的資料。

Parameters

- `sourceColumn` – 指定可能包含極端值的現有數值資料欄的名稱。
- `targetColumn` – 指定可能包含極端值的現有數值資料欄的名稱。
- `outlierStrategy` – 指定用於偵測極端值的方法。有效值包括下列項目：
  - `Z_SCORE` – 當值偏離平均值超過標準差閾值時，將值識別為極端值。

  - MODIFIED_Z_SCORE – 當值偏離中位數超過中位數絕對偏差閾值時，將值識別為極端值。
  - IQR – 當值超過欄資料的第一分位數和最後一分位數時，將值識別為極端值。四分位數範圍 (IQR) 測量資料點中間 50% 的位置。
- threshold – 指定偵測極端值時要使用的閾值。如果以 計算的分數outlierStrategy超過此數字，則sourceColumn值會識別為極端值。預設為 3。
- skewFunction – 指定取代極端值時要使用的方法。有效值包括下列項目：
  - LOG – 套用強大的轉換，以減少正面和負面扭曲。這是自然對數 (2.718281828)。
  - ROOT （使用 value = 3）– 套用相當強大的轉換，以減少正面和負面扭曲。（立方體根）
  - ROOT （使用 value = 2）– 套用中度轉換，僅減少正偏斜。（方形根）
  - SQUARE – 套用中度轉換以減少負偏斜。（方形）
  - 自訂轉換 – 使用 value 參數中提供的自訂編號套用指定的 LOG或 ROOT 轉換。
- value – 指定用於自訂轉換的值。如果 skewFunction是 LOG，則此值代表日誌的基礎。如果 skewFunction是 ROOT，則此值代表根的強大功能。

下列範例顯示單一RecipeAction操作的語法。配方至少包含一個RecipeStep操作，而配方步驟至少包含一個配方動作。配方動作會執行您指定的資料轉換。一組配方動作會依序執行，以建立最終資料集。

JSON

以下顯示使用 JSON 語法做為 RecipeStep DataBrew Recipe 範例RecipeAction成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example JSON 中的範例

```
{
    "Action": {
        "Operation": "RESCALE_OUTLIERS_WITH_SKEW",
        "Parameters": {
            "outlierStrategy": "Z_SCORE",
            "threshold": "3",
            "skewFunction": "ROOT",
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "value": "4"
        }
    }
}
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些和其他 API 操作。

YAML

以下顯示RecipeAction使用 YAML 語法做為 RecipeStep DataBrew Recipe 範例成員的範例。如需顯示配方動作清單的語法範例，請參閱 定義配方結構。

Example YAML 中的範例

```
- Action:
  Operation: RESCALE_OUTLIERS_WITH_SKEW
  Parameters:
    outlierStrategy: Z_SCORE
    threshold: '3'
    skewFunction: ROOT
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    value: '4'
```

如需在 API 操作中使用此配方動作的詳細資訊，請參閱 CreateRecipe或 UpdateRecipe。您可以在自己的程式碼中使用這些和其他 API 操作。

資料欄結構配方步驟

使用這些資料欄結構配方步驟來修改資料的資料欄結構。

主題

- BOOLEAN_OPERATION
- CASE_OPERATION
- FLAG_COLUMN_FROM_NULL
- FLAG_COLUMN_FROM_PATTERN
- MERGE
- SPLIT_COLUMN_BETWEEN_DELIMITER
- SPLIT_COLUMN_BETWEEN_POSITIONS
- SPLIT_COLUMN_FROM_END
- SPLIT_COLUMN_FROM_START

- SPLIT_COLUMN_MULTIPLE_DELIMITER
- SPLIT_COLUMN_SINGLE_DELIMITER
- SPLIT_COLUMN_WITH_INTERVALS

BOOLEAN_OPERATION

根據邏輯條件 IF 的結果建立新的資料欄。如果布林表達式為 true，則傳回 true 值；如果布林表達式為 false，則傳回 false 值，或傳回自訂值。

Parameters

- `trueValueExpression` – 符合條件時的結果。
- `falseValueExpression` – 不符合條件時的結果。
- `valueExpression` – 布林值條件。
- `withExpressions` – 彙總結果的組態。
- `targetColumn` – 新建立資料欄的名稱。

您可以在 trueValueExpression、falseValueExpression 和 valueExpression 中使用常數值、資料欄參考和彙總結果。

Example範例：常數值

保持不變的值，例如數字或句子。

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example範例：資料欄參考

資料集中資料欄的值。

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example範例：彙總結果

由彙總函數計算的值。彙總函數會對資料欄執行計算，並傳回單一值。

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`:mincolumn.2`",
        "falseValueExpression": "`:maxcolumn.3`",
        "valueExpression": "`column.1` < `:avgcolumn.4`",
        "withExpressions": "[{\"name\":\"mincolumn.2\",\"value\":\"min(`column.2`)\",
\"type\":\"aggregate\"},{\"name\":\"maxcolumn.3\",\"value\":\"max(`column.3`)\",\"type
\":\"aggregate\"},{\"name\":\"avgcolumn.4\",\"value\":\"avg(`column.4`)\",\"type\":
\"aggregate\"}]",
        "targetColumn": "result.column"
      }
    }
  }
}
```

使用者需要透過逸出將 JSON 轉換為字串。

請注意，trueValueExpression、falseValueExpression 和 valueExpression 中的參數名稱必須與 withExpressions 中的名稱相符。若要使用某些資料欄的彙總結果，您需要為其建立參數並提供彙總函數。

Example範例：

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example範例： 和/或

您可以使用 和 或 來合併多個條件。

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000 and `column.2` >= `column.3",
        "targetColumn": "result.column"
      }
    }
  }
}
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
```

```
        "trueValueExpression": "`column.4`",
        "falseValueExpression": "`column.5`",
        "valueExpression": "startsWith(`column1`, 'value1') or endsWith(`column2`,
 'value2')",
        "targetColumn": "result.column"
      }
    }
  }
}
```

有效的彙總函數

下表顯示可用於布林值操作的所有有效彙總函數。

資料欄類型	條件	valueExpression	withExpressions	傳回值
數值	總和	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	傳回 的總和 column.1
	Mean	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`col`	傳回 的平均值 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
			umn.1`)", "type": "aggregat e" }]	
	平均絕對偏差	`：meanabs olutedevi ation.column.1`	[{ "name": "meanabso lutedevia tion.colu mn.1", "value": "mean_abs olute_dev iation(`c olumn.1`) ", "type": "aggregat e" }]	傳回 的平均絕對偏差 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	Median	`：median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	傳回 的中位數 column.1
	產品	`：product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	傳回 的乘積 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	標準偏差	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	傳回 的標準差 `column.1`
	變異數	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	傳回 的變異數 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
	平均值的標準錯誤	`:standarderrorofmean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	傳回 平均值的標準錯誤 `column.1`
	偏斜	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	傳回 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
	峰度	`：kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	傳回 的峰度 `column.1`
Datetime/ Numeric/Text	計數	`：count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	傳回 中的資料列總數 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
	相異計數	`：countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	傳回 中不同資料列的總數 column.1
	最少	`：min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	傳回 的最小值 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	最多	`：max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	傳回 的最大值 column.1

valueExpression 中的有效條件

下表顯示支援的條件和您可以使用的值表達式。

資料欄類型	條件	valueExpression	Description
String	Contains	contains(`column`, 'text')	要測試資料欄中的值是否包含文字的條件
	不包含	！contains(`column`, 'text')	要測試資料欄中的值是否不包含文字的條件
	相符	matches(`column`, 'pattern')	要測試資料欄中的值是否符合模式的條件
	不符合	！matches(`column`, 'pattern')	要測試資料欄中的值是否與模式不相符的條件

資料欄類型	條件	valueExpression	Description
	開頭為	startsWith(`column`, 'text')	測試資料欄中的值是否以文字開頭的條件
	開頭不是	！startsWith(`column`，'text')	要測試資料欄中的值是否不是以文字開頭的條件
	Ends with	endsWith(`column`, 'text')	要測試資料欄中的值是否以文字結尾的條件
	結尾不是	！endsWith(`column`，'text')	要測試資料欄中的值是否以文字結尾的條件
數值	Less than	`column` < number	要測試資料欄中的值是否小於數字的條件
	小於或等於	`column` <= number	要測試資料欄中的值是否小於或等於數字的條件
	Greater than	`column` > 數字	要測試資料欄中的值是否大於數字的條件
	大於或等於	`column` >= number	要測試資料欄中的值是否大於或等於數字的條件
	介於	isBetween(`column`、minNumber、maxNumber)	要測試資料欄中的值是否介於 minNumber 和 maxNumber 之間的條件

資料欄類型	條件	valueExpression	Description
	不在 之間	！isBetween(`column`、minNumber、maxNumber)	要測試資料欄中的值是否不在 minNumber 和 maxNumber 之間的條件
Boolean	為 true	`column` = TRUE	要測試資料欄中的值是否為布林值 TRUE 的條件
	為 false	`column` = FALSE	要測試資料欄中的值是否為布林值 FALSE 的條件
日期/時間戳記	早於	`column` < 'date'	要測試資料欄中的值是否早於日期的條件
	早於或等於	`column` <= 'date'	要測試資料欄中的值是否早於或等於日期的條件
	晚於	`column` > 'date'	要測試資料欄中的值是否晚於日期的條件
	大於或等於	`column` >= 'date'	要測試資料欄中的值是否晚於或等於日期的條件
String/Numeric/Date/時間戳記	完全是	`column` = 'value'	測試資料欄中的值是否為確切值的條件
	Is not	`column` ！= 'value'	要測試資料欄中的值是否不是值的條件
	遺失	isMissing(`column`)	測試資料欄中的值是否遺失的條件

資料欄類型	條件	valueExpression	Description
	未遺失	！isMissing(`column`)	測試資料欄中的值是否未遺失的條件
	有效	isValid(`column`，資料類型）	測試資料欄中的值是否有效的條件（該值為資料類型，或可轉換為資料類型）
	無效	！isValid(`column`，資料類型）	測試資料欄中的值是否無效的條件（該值為資料類型，或可轉換為資料類型）
巢狀	遺失	isMissing(`column`)	測試資料欄中的值是否遺失的條件
	未遺失	！isMissing(`column`)	測試資料欄中的值是否未遺失的條件
	有效	isValid(`column`，資料類型）	測試資料欄中的值是否有效的條件（該值為資料類型，或可轉換為資料類型）
	無效	！isValid(`column`，資料類型）	測試資料欄中的值是否無效的條件（該值為資料類型，或可轉換為資料類型）

CASE_OPERATION

根據邏輯條件 CASE 的結果建立新的資料欄。案例操作會經歷案例條件，並在符合第一個條件時傳回值。一旦條件成立，操作就會停止讀取並傳回結果。如果沒有條件為 true，則會傳回預設值。

Parameters

- `valueExpression` – 條件。

- withExpressions – 彙總結果的組態。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "CASE_OPERATION",
      "Parameters": {
        "valueExpression": "case when `column11` < `column.2` then 'result1' when
 `column2` < 'value2' then 'result2' else 'high' end",
        "targetColumn": "result.column"
      }
    }
  }
}
```

有效的彙總函數

下表顯示可在案例操作中使用的所有有效彙總函數。

資料欄類型	條件	valueExpression	withExpressions	傳回值
數值	總和	`：sum.column.1`	[{ "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" }	傳回 的總和 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
			]	
	Mean	`：mean.column.1`	[{ "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" }]	傳回 的平均值 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	平均絕對偏差	`：meanabsolutedeviation.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	傳回 的平均絕對偏差 column.1
	Median	`：median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	傳回 的中位數 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	產品	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	傳回 的乘積 column.1
	標準偏差	`:standarddeviation.column.1`	`[ { "name": "standard deviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	傳回 的標準差 column.1

資料欄類型	條件	valueExpression	withExpressions	傳回值
	變異數	`：varianc e.column.1`	`[ { "name": "variance .column.1 ", "value": "variance (`column. 1`)", "type": "aggregat e" } ]`	傳回 的變異數 `column.1`
	平均值的標準錯誤	`：standar derrorofm ean.column.1`	`[ { "name": "standard errorofme an.column .1", "value": "standard _error_of _mean(`co lumn.1`)", "type": "aggregat e" } ]`	傳回 平均值的標準錯誤 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
	偏斜	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	傳回 `column.1`
	峰度	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	傳回 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
Datetime/ Numeric/Text	計數	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	傳回 中的資料列總數 `column.1`
	相異計數	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	傳回 中不同資料列的總數 `column.1`

資料欄類型	條件	valueExpression	withExpressions	傳回值
	最少	`：min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	傳回 的最小值 column.1
	最多	`：max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	傳回 的最大值 column.1

valueExpression 中的有效條件

下表顯示支援的條件和您可以使用的值表達式。

資料欄類型	條件	valueExpression	Description
String	Contains	contains(`column`, 'text')	要測試資料欄中的值是否包含文字的條件
	不包含	！contains(`column`, 'text')	要測試資料欄中的值是否不包含文字的條件
	相符	matches(`column`, 'pattern')	要測試資料欄中的值是否符合模式的條件
	不符合	！matches(`column`，'pattern')	要測試資料欄中的值是否與模式不相符的條件
	開頭為	startsWith(`column`, 'text')	測試資料欄中的值是否以文字開頭的條件
	開頭不是	！startsWith(`column`，'text')	要測試資料欄中的值是否不是以文字開頭的條件
	Ends with	endsWith(`column`, 'text')	要測試資料欄中的值是否以文字結尾的條件
	結尾不是	！endsWith(`column`，'text')	要測試資料欄中的值是否以文字結尾的條件
數值	Less than	`column` < number	要測試資料欄中的值是否小於數字的條件
	小於或等於	`column` <= number	要測試資料欄中的值是否小於或等於數字的條件

資料欄類型	條件	valueExpression	Description
	Greater than	`column` > 數字	要測試資料欄中的值是否大於數字的條件
	大於或等於	`column` >= number	要測試資料欄中的值是否大於或等於數字的條件
	介於	isBetween(`column`，minNumber，maxNumber)	要測試資料欄中的值是否介於 minNumber 和 maxNumber 之間的條件
	不在 之間	！isBetween(`column`，minNumber，maxNumber)	要測試資料欄中的值是否不在 minNumber 和 maxNumber 之間的條件
Boolean	為 true	`column` = TRUE	要測試資料欄中的值是否為布林值 TRUE 的條件
	為 false	`column` = FALSE	要測試資料欄中的值是否為布林值 FALSE 的條件
日期/時間戳記	早於	`column` < 'date'	要測試資料欄中的值是否早於日期的條件
	早於或等於	`column` <= 'date'	要測試資料欄中的值是否早於或等於日期的條件
	晚於	`column` > 'date'	要測試資料欄中的值是否晚於日期的條件

資料欄類型	條件	valueExpression	Description
	大於或等於	`column` >= 'date'	要測試資料欄中的值是否晚於或等於日期的條件
String/Numeric/Date/時間戳記	完全是	`column` = 'value'	測試資料欄中的值是否為確切值的條件
	Is not	`column` ！= 'value'	要測試資料欄中的值是否不是值的條件
	遺失	isMissing(`column`)	測試資料欄中的值是否遺失的條件
	未遺失	！isMissing(`column`)	測試資料欄中的值是否未遺失的條件
	有效	isValid(`column`，資料類型）	測試資料欄中的值是否有效的條件（該值為資料類型，或可轉換為資料類型）
	無效	！isValid(`column`，資料類型）	測試資料欄中的值是否無效的條件（該值為資料類型，或可轉換為資料類型）
巢狀	遺失	isMissing(`column`)	測試資料欄中的值是否遺失的條件
	未遺失	！isMissing(`column`)	測試資料欄中的值是否未遺失的條件
	有效	isValid(`column`，資料類型）	測試資料欄中的值是否有效的條件（該值為資料類型，或可轉換為資料類型）

資料欄類型	條件	valueExpression	Description
	無效	！isValid(`column`，資料類型）	測試資料欄中的值是否無效的條件（該值為資料類型，或可轉換為資料類型）

FLAG_COLUMN_FROM_NULL

根據現有資料欄中存在的 null 值建立新的資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄名稱。
- `flagType` – 必須設定為 的值`Null values`。
- `trueString` – 如果在來源中找到 null 值，則為新資料欄的值。如未指定任何值，預設為 `True`。
- `falseString` – 如果在來源中找到非空值，則為新資料欄的值。如未指定任何值，預設為 `False`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

根據現有資料欄中使用者指定的模式，建立新的資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄名稱。
- flagType – 必須設定為 的值Pattern。
- pattern – 規則表達式，指出要評估的模式。
- trueString – 如果在來源中找到 null 值，則為新資料欄的值。如未指定任何值，預設為 True。
- falseString – 如果在來源中找到非空值，則為新資料欄的值。如未指定任何值，預設為 False。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
    }
}
```

MERGE

將兩個或多個資料欄合併到新的資料欄。

Parameters

- sourceColumns – JSON 編碼字串，代表要合併的一或多個資料欄清單。
- delimiter – 要在目標欄中顯示的值之間的選用分隔符號。
- targetColumn – 要建立的合併資料欄名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_DELIMITER

根據開始和結束分隔符號，將資料欄分割為三個新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `patternOption1` – JSON 編碼字串，代表表示第一個分隔符號的一或多個字元。
- `patternOption2` – JSON 編碼字串，代表表示第二個分隔符號的一或多個字元。
- `pattern` – 分割資料時要用作分隔符號的一個或多個字元。
- `includeInSplit` – 如果為 true，請在新資料欄中包含模式；否則會捨棄模式。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_DELIMITER",
        "Parameters": {
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_POSITIONS

根據您指定的位移，將資料欄分割成三個新的資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- startPosition – 要開始分割的字元位置。
- endPosition – 分割要結束的字元位置。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
        }
    }
}
```

SPLIT_COLUMN_FROM_END

將資料欄分割成兩個新資料欄，與字串結尾位移。

Parameters

- sourceColumn – 現有資料欄的名稱。
- position – 字元位置，從字串的右端開始，將發生分割。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
```

```
}
```

SPLIT_COLUMN_FROM_START

將資料欄分割成兩個新的資料欄，與字串開頭的位移。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `position` – 字元位置，從字串的左端開始，將發生分割。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

SPLIT_COLUMN_MULTIPLE_DELIMITER

根據多個分隔符號分割資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `patternOptions` – JSON 編碼字串，代表決定分割條件的一或多個模式。
- `pattern` – 分割資料時要用作分隔符號的一個或多個字元。
- `limit` – 要執行的分割數量。最小值為 1；最大值為 20。
- `includeInSplit` – 如果為 true，請在新資料欄中包含模式；否則會捨棄模式。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
}
```

SPLIT_COLUMN_SINGLE_DELIMITER

根據特定分隔符號，將資料欄分割為一或多個新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `pattern` – 分割資料時要用作分隔符號的一個或多個字元。
- `limit` – 要執行的分割數量。最小值為 1；最大值為 20。
- `includeInSplit` – 如果為 true，請在新資料欄中包含模式；否則會捨棄模式。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_SINGLE_DELIMITER",
        "Parameters": {
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVALS

以 n 個字元的間隔分割資料欄，您可以在其中指定 n。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `startPosition` – 要開始分割的字元位置。
- `interval` – 在下一個分割之前要略過的字元數。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

資料欄格式化配方步驟

使用資料欄格式化配方步驟來變更資料欄中資料的格式。

主題

- NUMBER_FORMAT
- FORMAT_PHONE_NUMBER

NUMBER_FORMAT

傳回將數值轉換為格式化字串的資料欄。

Parameters

- `sourceColumn` – 字串. 現有資料欄的名稱。
- `decimalPlaces` – 整數。十進位分隔符號後位數的值。
- `numericDecimalSeparator` – 字串. 下列其中一個值表示小數分隔符號：
  - "."

  - ","
- numericThousandSeparator – 字串. 下列其中一個值表示千個分隔符號：
  - null。表示未啟用千個分隔符號。
  - ","
  - " "
  - "."
  - "\\"
- numericAbbreviatedUnit – 字串. 下列其中一個值表示縮寫單位：
  - null。表示未啟用縮寫單位。
  - 「THOUSAND」
  - 「百萬」
  - 「十億」
  - "TRILLION"
- numericUnitAbbreviation – 字串. 下列其中一個值或任何自訂值，表示單位縮寫：
  - null。表示未啟用單位縮寫。
  -

縮寫單位	選項
數千	K、k、M、千、自訂
百萬	M、m、MM、百萬、自訂
十億	B、bn、十億、自訂
兆	T、tn、兆、自訂

Example範例

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
```

```
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
            "numericUnitAbbreviation": "K"
        }
    }
}
```

FORMAT_PHONE_NUMBER

傳回將電話號碼字串轉換為格式化值的資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- phoneNumberFormat – 將電話號碼轉換為的格式。如果未指定格式，預設值為 E.164，這是國際認可的標準電話號碼格式。有效值包括下列項目：
  - E164 （省略 之後的期間E)
- defaultRegion – 由兩個或三個大寫字母組成的有效區域代碼，當號碼本身沒有國家/地區代碼時，用於指定電話號碼所在的區域。最多可以提供 defaultRegion 或 defaultRegionColumn 之一。
- defaultRegionColumn – 進階資料類型 的資料欄名稱Country。當號碼本身沒有國家/地區代碼時，所指定資料欄中的區域代碼用於確定電話號碼的國家/地區代碼。最多可以提供 defaultRegion 或 defaultRegionColumn 之一。

備註

- 無法格式化為有效電話號碼的輸入保持不變。
- 如果未提供預設區域，且電話號碼開頭不是加號 (+) 和國碼，則電話號碼不會格式化。

Example

範例：固定預設區域

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
```

```
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

範例：預設區域資料欄選項

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

資料結構配方步驟

使用這些配方步驟，從不同的角度製表和摘要資料，或執行進階函數。

主題

- NEST_TO_ARRAY
- NEST_TO_MAP
- NEST_TO_STRUCT
- UNNEST_ARRAY
- UNNEST_MAP
- UNNEST_STRUCT
- UNNEST_STRUCT_N
- GROUP_BY
- JOIN
- PIVOT
- SCALE

- 轉置
- UNION
- UNPIVOT

NEST_TO_ARRAY

將使用者選取的資料欄轉換為陣列值。建立結果陣列時，會維持所選資料欄的順序。不同的資料欄資料類型是類型轉換為支援所有資料欄資料類型的常見類型。

Parameters

- `sourceColumns` — 來源資料欄的清單。
- `targetColumn` — 目標欄的名稱。
- `removeSourceColumns` — 包含 `true`或 值`false`，指出使用者是否想要移除選取的來源資料欄。

Example範例

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_MAP

將使用者選取的資料欄轉換為鍵值對，每個資料欄都有代表資料欄名稱的索引鍵，以及代表資料列值的值。建立結果映射時，不會維持所選資料欄的順序。不同的資料欄資料類型是類型轉換為支援所有資料欄資料類型的常見類型。

Parameters

- `sourceColumns` — 來源資料欄的清單。

- targetColumn — 目標欄的名稱。
- removeSourceColumns — 包含 true或 值false，指出使用者是否想要移除選取的來源資料欄。

Example範例

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_STRUCT

將使用者選取的資料欄轉換為鍵值對，每個資料欄都有代表資料欄名稱的索引鍵，以及代表資料列值的值。所選資料欄的順序和每個資料欄的資料類型會保留在產生的結構中。

Parameters

- sourceColumns — 來源資料欄的清單。
- targetColumn — 目標欄的名稱。
- removeSourceColumns — 包含 true或 值false，指出使用者是否想要移除選取的來源資料欄。

Example範例

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
```

```
    }
}
```

UNNEST_ARRAY

將 類型的資料欄取消巢狀array化為新的資料欄。如果陣列包含多個值，則會產生對應至每個元素的資料列。此函數只會將陣列資料欄的一個層級解除巢狀化。

Parameters

- sourceColumn — 現有資料欄的名稱。此欄必須為 struct類型。
- targetColumn — 產生的目標資料欄名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_MAP

將 類型的資料欄解除巢狀化，map並產生索引鍵和值的資料欄。如果有多個索引鍵/值對，則會產生對應至每個索引鍵值的資料列。此函數只會將映射資料欄的一個層級解除巢狀化。

Parameters

- sourceColumn — 現有資料欄的名稱。此欄必須為 struct類型。
- removeSourceColumn — 如果為 true，來源資料欄會在函數完成後刪除。
- targetColumn — 如果提供，則每個產生的資料欄都會以此字首開頭。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

將 類型的資料欄取消巢狀化，struct並為結構中存在的每個索引鍵產生資料欄。此函數只會解除巢狀結構層級 1。

Parameters

- sourceColumn — 現有資料欄的名稱。此欄必須為結構類型。
- removeSourceColumn — 如果為 true，來源資料欄會在函數完成後刪除。
- targetColumn — 如果提供，則每個產生的資料欄都會以此字首開頭。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

為類型為 的所選資料欄的每個欄位建立新的資料欄struct。

例如，指定下列結構：

```
            user {
               name: “Ammy”
               address: {
                  state: "CA",
                  zipcode: 12345
               }
            }
```

此函數會建立 3 個資料欄：

user.name	user.address.state	user.address.zipcode
Ammy	CA	12345

Parameters

- `sourceColumns` — 來源資料欄的清單。
- `regexColumnSelector` — 用於選取要取消巢狀的資料欄的規則表達式。
- `removeSourceColumn` — 布林值。如果為 true，請移除來源資料欄，否則請保留。
- `unnestLevel` — 要取消巢狀化的關卡數量。
- `delimiter` — 分隔符號用於新建立的資料欄名稱，以分隔結構的不同層級。例如：如果分隔符號為 “/”，資料欄名稱將採用以下格式：“user/address/state”。
- `conditionExpressions` — 條件表達式。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT_N",
        "Parameters": {
            "sourceColumns": "[\"address\"]",
            "removeSourceColumn": "true",
```

```
            "unnestLevel": "2",
            "delimiter": "/"
        }
    }
}
```

GROUP_BY

透過依一或多個資料欄分組資料列，然後將彙總函數套用至每個群組，來摘要資料。

Parameters

- `sourceColumns` — JSON 編碼字串，代表構成每個群組基礎的資料欄清單。
- `groupByAggFunctions` — JSON 編碼字串，代表要套用的彙總函數清單。（如果您不想彙總，請指定 UNAGGREGATED。）
- `useNewDataFrame` — 如果為 true，則 GROUP_BY 的結果會在專案工作階段中提供，取代其目前的內容。

Example範例

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
      }
    }
  }
]
```

JOIN

在兩個資料集上執行聯結操作。

Parameters

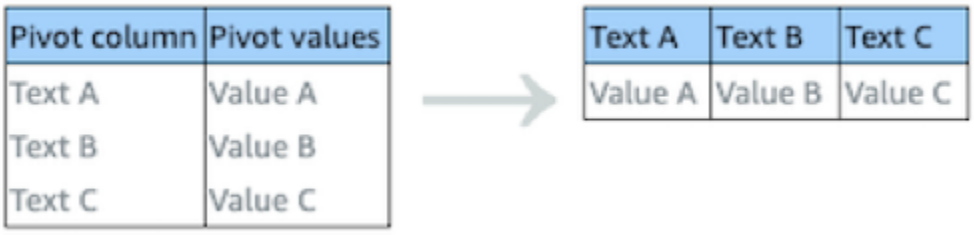
- joinKeys — JSON 編碼字串，代表每個資料集的資料欄清單，做為聯結金鑰。
- joinType — 要執行的聯結類型。必須是其中之一： INNER_JOIN | LEFT_JOIN | RIGHT_JOIN | OUTER_JOIN | LEFT_EXCLUDING_JOIN | | RIGHT_EXCLUDING_JOIN | OUTER_EXCLUDING_JOIN
- leftColumns — JSON 編碼字串，代表目前作用中資料集的資料欄清單。
- rightColumns — JSON 編碼字串，代表從另一個 （次要） 資料集加入目前資料集的資料欄清單。
- secondInputLocation — Amazon S3 URL，可解析為次要資料集的資料檔案。
- secondaryDatasetName — 次要資料集的名稱。

Example範例

```
{
    "Action": {
        "Operation": "JOIN",
        "Parameters": {
            "joinKeys": "[{\"key\":\"assembly_session\",\"value\":\"assembly_session
\"},{\"key\":\"state_code\",\"value\":\"state_code\"}]",
            "joinType": "INNER_JOIN",
            "leftColumns": "[\"year\",\"assembly_session\",\"state_code\",\"state_name
\",\"all_votes\",\"yes_votes\",\"no_votes\",\"abstain\",\"idealpoint_estimate
\",\"affinityscore_usa\",\"affinityscore_russia\",\"affinityscore_china\",
\"affinityscore_india\",\"affinityscore_brazil\",\"affinityscore_israel\"]",
            "rightColumns": "[\"assembly_session\",\"vote_id\",\"resolution\",
\"state_code\",\"state_name\",\"member\",\"vote\"]",
            "secondInputLocation": "s3://databrew-public-datasets-us-east-1/votes.csv",
            "secondaryDatasetName": "votes"
        }
    }
}
```

PIVOT

將所選資料欄中的所有資料列值轉換為具有值的個別資料欄。



Parameters

- sourceColumn — 現有資料欄的名稱。資料欄最多可有 10 個不同的值。
- valueColumn — 現有資料欄的名稱。資料欄最多可有 10 個不同的值。
- aggregateFunction — 彙總函數的名稱。如果您不想要彙總，請使用關鍵字 COLLECT_LIST。

Example範例

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
            "valueColumn": "all_votes"
        }
    }
}
```

SCALE

擴展或標準化數值欄中的資料範圍。

Parameters

- sourceColumn — 現有資料欄的名稱。
- strategy — 要套用至資料欄值的操作：
  - MIN_MAX — 將值重新擴展到 【0，1】 的範圍。
  - SCALE_BETWEEN — 將值重新擴展為兩個指定值的範圍。
  - MEAN_NORMALIZATION — 重新調整資料規模，使其平均值 (μ) 為 0，標準差 (σ) 為 1，範圍為 【-1， 1】。

  - Z_SCORE — 線性擴展資料值，使平均值 (μ) 為 0，標準差 (σ) 為 1。最適合處理極端值。
- targetColumn — 要包含結果的資料欄名稱。

Example範例

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

轉置

將所有選取的資料列轉換為資料欄，將資料欄轉換為資料列。

Column 1	Column A	Column B	Column C
Row A	Value A	Value B	Value C
Row B	Value A1	Value B1	Value C1

New column	Row A	Row B
Column A	Value A	Value A1
Column B	Value B	Value B1
Column C	Value C	Value C1

Parameters

- pivotColumns — JSON 編碼字串，代表資料列將轉換為資料欄名稱的資料欄清單。
- valueColumns — JSON 編碼字串，代表要轉換為資料列的一或多個資料欄清單。
- aggregateFunction — 彙總函數的名稱。如果您不想要彙總，請使用關鍵字 COLLECT_LIST。
- newColumn — 將轉置的資料欄保留為值的資料欄。

Example範例

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
            "newColumn": "Student"
        }
    }

}
```

UNION

將來自兩個或多個資料集的資料列合併為單一結果。

Parameters

- `datasetsColumns` — JSON 編碼字串，代表資料集中所有資料欄的清單。
- `secondaryDatasetNames` — JSON 編碼字串，代表一或多個次要資料集的清單。
- `secondaryInputs` — JSON 編碼字串，代表 Amazon S3 儲存貯體和物件金鑰名稱的清單，告知 DataBrew 尋找次要資料集的位置 (s)。
- `targetColumnNames` — JSON 編碼字串，代表結果的資料欄名稱清單。

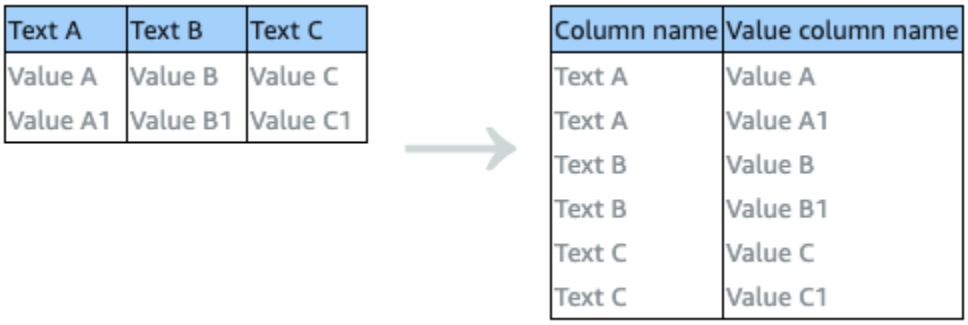
Example範例

```
{
    "Action": {
        "Operation": "UNION",
        "Parameters": {
            "datasetsColumns": "[[\"assembly_session\",\"state_code\",
\"state_name\",\"year\",\"all_votes\",\"yes_votes\",\"no_votes\",\"abstain
\",\"idealpoint_estimate\",\"affinityscore_usa\",\"affinityscore_russia\",
\"affinityscore_china\",\"affinityscore_india\",\"affinityscore_brazil\",
\"affinityscore_israel\"],[\"assembly_session\",\"state_code\",\"state_name
\",null,null,null,null,null,null,null,null,null,null,null,null]]",
```

```
            "secondaryDatasetNames": "[\"votes\"]",
            "secondaryInputs": "[{\"S3InputDefinition\":{\"Bucket\":\"databrew-public-
datasets-us-east-1\",\"Key\":\"votes.csv\"}}]",
            "targetColumnNames": "[\"assembly_session\",\"state_code\",\"state_name\",
\"year\",\"all_votes\",\"yes_votes\",\"no_votes\",\"abstain\",\"idealpoint_estimate
\",\"affinityscore_usa\",\"affinityscore_russia\",\"affinityscore_china\",
\"affinityscore_india\",\"affinityscore_brazil\",\"affinityscore_israel\"]"
        }
    }
}
```

UNPIVOT

將所選資料列中的所有資料欄值轉換為具有值的個別資料列。



Text A	Text B	Text C
Value A	Value B	Value C
Value A1	Value B1	Value C1

Column name	Value column name
Text A	Value A
Text A	Value A1
Text B	Value B
Text B	Value B1
Text C	Value C
Text C	Value C1

Parameters

- `sourceColumns` — JSON 編碼字串，代表要撤銷的一或多個資料欄清單。
- `unpivotColumn` — 取消樞紐分析操作的值欄。
- `valueColumn` — 要保留已撤銷值的資料欄。

Example範例

```
{
    "Action": {
        "Operation": "UNPIVOT",
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
```

```
    }
}
```

資料科學配方步驟

使用這些配方步驟，從不同的角度製表和摘要資料，或執行進階轉換。

主題

BINARIZATION

取得所選數值來源資料欄中的所有值，將它們與閾值進行比較，並為每個資料列輸出具有 1 或 0 的新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

  `targetColumn` – 要建立的新資料欄的名稱。

  `threshold` – 指出指派值為 0 或 1 之閾值的數字。

  `flip` – 翻轉二進位指派的選項，以便指派較低的值 1，而較高的值則指派 0。當翻轉參數為 true 時，小於或等於閾值的值會產生 1，而大於閾值的值會產生 0。

Example範例

```
{
    "Action": {
```

```
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
}
```

儲存貯體化

儲存貯體 （在主控台中稱為 Binning) 會取得數值資料欄中的項目，將它們分組為數值範圍定義的儲存貯體，並輸出顯示每一列之儲存貯體的新資料欄。您可以使用分割或百分比完成歸納。以下第一個範例使用分割，第二個範例使用百分比。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

  `targetColumn` – 要建立的新資料欄的名稱。

  `bucketNames` – 儲存貯體名稱的清單。

  `splits` – 儲存貯體層級的清單。儲存貯體是連續的，而儲存貯體的上限將是下一個儲存貯體的下限。

  `percentage` – 每個儲存貯體將以百分比描述。

Example使用分割的範例

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
        }
    }
```

```
}
```

Example使用百分比的範例

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

CATEGORICAL_MAPPING

將一或多個分類值映射至數值或其他值

Parameters

- `sourceColumn` – 現有資料欄的名稱。

  `categoryMap` – JSON 編碼字串，代表將值映射到類別。

  `deleteOtherRows` – 如果為 `true`，則會從資料集中移除所有未映射的資料列。

  `other` – 提供時，所有未映射的值都會被此值取代。

  `keepOthers` – 如果為 true，所有未映射的值將保持不變。

  `mapType` – 映射資料欄的資料類型。

  `targetColumn` – 要包含結果的資料欄名稱。

Example範例

```
{
    "Action": {
        "Operation": "CATEGORICAL_MAPPING",
```

```
        "Parameters": {
            "categoryMap": "{\"United States of America\":\"1\",\"Canada\":\"2\",\"Cuba
\":\"3\",\"Haiti\":\"4\",\"Dominican Republic\":\"5\"}",
            "deleteOtherRows": "false",
            "keepOthers": "true",
            "mapType": "NUMERIC",
            "sourceColumn": "state_name",
            "targetColumn": "state_name_mapped"
        }
    }
}
```

ONE_HOT_ENCODING

建立 n 個數值欄，其中 n 是所選分類變數中唯一值的數目。

例如，請考慮名為 的資料欄shirt_size。襯衫提供小型、中型、大型或超大型。資料欄資料可能如下所示。

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
M
```

在此案例中， 有四個不同的值shirt_size。因此， ONE_HOT_ENCODING會產生四個新的資料欄。每個新資料欄都名為 shirt_size_*x*，其中 *x*代表不同的shirt_size 值。

shirt_size 和四個產生的資料欄的結果如下所示。

```
shirt_size    shirt_size_S    shirt_size_M    shirt_size_L    shirt_size_XL
------------    ------------    ------------    ------------    -------------
L              0               0               1               0
```

```
XL              0               0               0               1
M               0               1               0               0
S               1               0               0               0
M               0               1               0               0
M               0               1               0               0
S               1               0               0               0
XL              0               0               0               1
M               0               1               0               0
L               0               0               1               0
XL              0               0               0               1
M               0               1               0               0
```

您為 指定的資料欄最多ONE_HOT_ENCODING可以有十 (10) 個不同的值。

Parameters

- sourceColumn – 現有資料欄的名稱。資料欄最多可有 10 個不同的值。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SCALE

擴展或標準化數值欄中的資料範圍。

Parameters

- sourceColumn – 現有資料欄的名稱。
- strategy – 要套用至資料欄值的操作：
  - MIN_MAX – 將值重新擴展到 【0，1】 的範圍
  - SCALE_BETWEEN – 將值重新擴展為 2 個指定值的範圍。

  - MEAN_NORMALIZATION – 重新調整資料規模，使其平均值 (μ) 為 0，標準差 (σ) 為 1，範圍為【-1， 1】
  - Z_SCORE – 線性擴展資料值，使平均值 (μ) 為 0，標準差 (σ) 為 1。最適合處理極端值。
- targetColumn – 要包含結果的資料欄名稱。

Example範例

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

偏斜

在資料值上套用轉換，以變更分佈形狀及其偏斜。

Parameters

- sourceColumn – 現有資料欄的名稱。

  targetColumn – 要建立的新資料欄的名稱。

  skewFunction
  - ROOT – 擷取 value-root。根可在 value 參數中提供。

    LOG – 日誌基本值。日誌庫可在 value 參數中提供。

    SQUARE – 平方函數

  value – skewFunction 的引數。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

字符化

將文字分割成較小的單位或字符，例如個別單字或詞彙。

Parameters

- sourceColumn – 現有資料欄的名稱。
- delimiter — 出現在字符化單字之間的自訂分隔符號。（預設行為是以空格分隔每個字符。）
- expandContractions — 如果為 ENABLED， 會展開合約單字。例如："don't" 會變成 "do not"。
- stemmingMode — 將文字分割成較小的單位或字符，例如個別小寫單字或詞彙。有兩種主幹模式可用： PORTER | LANCASTER。
- stopWordRemovalMode — 移除常見字詞，例如 a、、 等。
- customStopWords — 針對 StopWordRemovalMode， 可讓您指定停止單字的自訂清單。
- targetColumn — 要包含結果的資料欄名稱。

Example範例

```
{
    "Action": {
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
```

```
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

數學函式

接下來，尋找適用於配方動作之數學函數的參考主題。

主題

- ABSOLUTE
- ADD
- CEILING
- DEGREES
- 分割
- 指數
- FLOOR
- IS_EVEN
- IS_ODD
- LN
- LOG
- MOD
- 乘以
- 否定
- PI
- POWER
- RADIANS
- RANDOM
- RANDOM_BETWEEN
- ROUND

- SIGN
- SQUARE_ROOT
- 減去

ABSOLUTE

傳回新資料欄中輸入號碼的絕對值。絕對值是數字從零開始的距離，無論是正值還是負值

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
        }
    }
}
```

ADD

使用 (`sourceColumn1` + `sourceColumn2`) 或 (`sourceColumn1` +) ，在新資料欄中摘要輸入資料欄值`value1`。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `value1` – 數值。
- `sourceColumn2` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

CEILING

傳回大於或等於新資料欄中輸入十進位數字的最小整數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value1` – 數值。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "CEILING",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

DEGREES

將角度的弧度轉換為度數，並在新的資料欄中傳回結果。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

分割

將一個輸入號碼除以另一個輸入號碼，並在新的資料欄中傳回結果。

Parameters

- sourceColumn1 – 現有資料欄的名稱。
- value1 – 數值。
- sourceColumn2 – 現有資料欄的名稱。
- value2 – 數值。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
```

```
        }
    }
}
```

指數

傳回 Euler 在新資料欄中增加到第 n 度的數字。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_EXPONENT"
        }
    }
}
```

FLOOR

傳回大於或等於新資料欄中輸入數字的最大整數。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `value` – 數值。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

IS_EVEN

傳回新資料欄中的布林值，指出來源資料欄或值是否均勻。如果來源資料欄或值為小數，則結果為 false。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。
- `trueString` – 指示值是否為偶數的字串。
- `falseString` – 指出值是否不均勻的字串。

Example範例

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
        "Parameters": {
            "falseString": "Value is odd",
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

IS_ODD

傳回新資料欄中的布林值，指出來源資料欄或值是否為奇數。如果來源資料欄或值為小數，則結果為 false。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。
- `trueString` – 指出值是否為奇數的字串。
- `falseString` – 指出值是否為奇數的字串。

Example範例

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
}
```

LN

傳回新資料欄中值的自然對數 (Euler 的數字）。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
```

```
        }
    }
}
```

LOG

傳回新資料欄中值的對數。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄的名稱。
- base – 對數的基礎。預設為 10。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LOG",
        "Parameters": {
            "base": "10",
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
    }
}
```

MOD

傳回新資料欄中一個數字為另一個數字的百分比。

Parameters

- sourceColumn1 – 現有資料欄的名稱。
- sourceColumn2 – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

乘以

將兩個數字相乘，並在新資料欄中傳回結果。

Parameters

- sourceColumn1 – 現有資料欄的名稱。
- value1 – 數值。
- sourceColumn2 – 現有資料欄的名稱。
- value2 – 數值。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

否定

否定值，並在新資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

傳回新資料欄中 pi (3.141592653589793) 的值。

Parameters

- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

POWER

將數字的值傳回至新資料欄中指數的度數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要提高其值的數字。
- `targetColumn` – 要建立的新資料欄的名稱。
- `exponent` – 將提高值的功率。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "POWER",
        "Parameters": {
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
    }
}
```

RADIANS

將度數轉換為弧度（除以 180/pi），並在新資料欄中傳回值。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

RANDOM

在新資料欄中傳回介於 0 到 1 之間的隨機數字。

Parameters

- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "RANDOM",
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

RANDOM_BETWEEN

在新資料欄中，會在指定的下限（包含）和指定的上限（包含）之間傳回隨機數字。

Parameters

- `lowerBound` – 隨機數字範圍的下限。
- `upperBound` – 隨機數字範圍的上限。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

ROUND

將數值四捨五入到新資料欄中最接近的整數。當分數為 0.5 或更高時，它會四捨五入。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

SIGN

如果值小於 0，則傳回含 -1 的新資料欄；如果值為 0，則傳回 0；如果值大於 0，則傳回 +1。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

SQUARE_ROOT

傳回新資料欄中值的平方根。

Parameters

- sourceColumn – 現有資料欄的名稱。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

減去

從另一個欄位減去一個數字，並在新資料欄中傳回結果。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `value1` – 數值。
- `sourceColumn2` – 現有資料欄的名稱。
- `value2` – 數值。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "targetColumn": "weight_minus_10_kg",
            "value2": "10"
        }
    }
}
```

彙總函數

接下來，尋找使用配方動作之彙總函數的參考主題。

主題

- ANY
- AVERAGE
- COUNT
- COUNT_DISTINCT
- KTH_LARGEST
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN

- MODE
- STANDARD_DEVIATION
- SUM
- VARIANCE

ANY

傳回新資料欄中所選來源資料欄的任何值。會忽略空值和 null 值。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

AVERAGE

計算來源資料欄中值的平均值，並在新資料欄中傳回結果。忽略任何非數字。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

COUNT

傳回新資料欄中所選來源資料欄的值數目。會忽略空值和 null 值。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

COUNT_DISTINCT

傳回新資料欄中所選來源資料欄的不同值總數。會忽略空值和 null 值。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "COUNT_DISTINCT",
        "Parameters": {
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
    }
}
```

KTH_LARGEST

從新資料欄中選取的來源資料欄傳回第 k 個最大數字。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。
- `value` – 代表 k 的數字。

Example範例

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

從新資料欄中選取的來源資料欄傳回第 k 個最大的唯一數字。

Parameters

- sourceColumns – JSON 編碼字串，代表現有資料欄的清單。
- targetColumn – 新建立資料欄的名稱。

  value – 代表 k 的數字。

Example範例

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

從新資料欄中選取的來源資料欄傳回數值上限。忽略任何非數字。

Parameters

- sourceColumns – JSON 編碼字串，代表現有資料欄的清單。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
```

```
}
```

MEDIAN

從新資料欄中選取的來源資料欄傳回數字群組排序的中位數、中間數。忽略任何非數字。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

從新資料欄中選取的來源資料欄傳回最小值。忽略任何非數字。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MIN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
```

```
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

從新資料欄中選取的來源資料欄傳回最常出現的模式數字。忽略任何非數字。對於多種模式，會使用模態函數計算模式。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

STANDARD_DEVIATION

傳回新資料欄中所選來源資料欄的標準差。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

SUM

傳回新資料欄中所選來源資料欄的值總和。任何非數字都會視為 0。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
    }
}
```

VARIANCE

傳回新資料欄中所選來源資料欄的差異。變異定義為 `Var(X) = [Sum ((X – mean(X))^2)]/ Count(X)`。

Parameters

- `sourceColumns` – JSON 編碼字串，代表現有資料欄的清單。

- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "VARIANCE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

文字函數

接下來，尋找使用配方動作之文字函數的參考主題。

主題

- CHAR
- ENDS_WITH
- 確切的
- 尋找
- LEFT
- LEN
- LOWER
- MERGE_COLUMNS_AND_VALUES
- 適當
- REMOVE_SYMBOLS
- REMOVE_WHITESPACE
- REPEAT_STRING
- RIGHT
- RIGHT_FIND
- STARTS_WITH

- STRING_GREATER_THAN
- STRING_GREATER_THAN_EQUAL
- STRING_LESS_THAN
- STRING_LESS_THAN_EQUAL
- SUBSTRING
- TRIM
- UNICODE
- UPPER

CHAR

在新資料欄中傳回來源資料欄中每個整數的 Unicode 字元，或傳回自訂整數值的 Unicode 字元。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 代表 Unicode 值的整數。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
}
```

ENDS_WITH

如果指定數目的最右側字元或自訂字串符合模式，`true`則會在新資料欄中傳回 。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `pattern` – 必須與字串結尾相符的規則運算式。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

確切的

建立填入下列其中一項的新資料欄：

- `True` 如果資料欄（或值）中的一個字串完全符合不同資料欄（或值）中的另一個字串。
- `False` 如果沒有相符項目。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - 兩者都是 `sourceColumn`*N*。
> - 其中一個 `sourceColumn`*N*和其中一個 `value`*N*。
> - 兩者都是 `value`*N*。

Example範例

```
{
    "RecipeAction": {
        "Operation": "EXACT",
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

尋找

由左至右搜尋，從來源資料欄或自訂值尋找符合指定字串的字串，並在新資料欄傳回結果。

Parameters

- sourceColumn – 現有資料欄的名稱。
- pattern – 要搜尋的規則表達式。
- position – 從字串左端開始的字元位置。
- ignoreCase – 如果為 true，請忽略字母之間的大小寫差異 （大小寫之間）。若要強制執行嚴格的比對，請false改用 。
- targetColumn – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

LEFT

根據字元數， 會從來源資料欄或自訂字串取得字串中最左邊的字元數，並傳回新資料欄中最左邊的指定字元數。

Parameters

- sourceColumn – 現有資料欄的名稱。
- value – 要評估的字元字串。
- position – 從字串左端開始的字元位置。

- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LEFT",
        "Parameters": {
            "position": "3",
            "sourceColumn": "city",
            "targetColumn": "city_left"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEFT",
        "Parameters": {
            "position": "5",
            "value": "How now brown cow",
            "targetColumn": "how_now_5_left_chars"
        }
    }
}
```

LEN

在新資料欄中傳回來源資料欄或自訂字串的字串長度。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
        }
    }
}
```

LOWER

將來源資料欄或自訂字串中的所有字母字元轉換為小寫，並在新的資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。

- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
    }
}
```

MERGE_COLUMNS_AND_VALUES

串連來源資料欄中的字串，並在新的資料欄中傳回結果。您可以在合併的值之間插入分隔符號。

Parameters

- `sourceColumns` – 兩個或多個現有資料欄的名稱，採用 JSON 編碼格式。
- `delimiter` - 選用。要在每個兩個來源資料欄值之間放置的一或多個字元。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
    }
}
```

適當

將來源資料欄或自訂值中字串的所有字母字元轉換為適當的大小寫，並在新的資料欄中傳回結果。

在適當情況下，也稱為大寫，每個單字的第一個字母會大寫，而該單字的其餘部分則會轉換為小寫。例如：Quick Brown Fox 跳過圍欄

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "sourceColumn": "first_name",
```

```
            "targetColumn": "first_name_proper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "value": "MR. H. SMITH, ESQ.",
            "targetColumn": "formal_name_proper"
        }
    }
}
```

REMOVE_SYMBOLS

從來源欄或自訂字串中的字串中移除非字母、數字、重音拉丁字元或空格的字元，並在新欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "sourceColumn": "info_url",
```

```
            "targetColumn": "info_url_remove_symbols"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "value": "$&#$&HEY!#@@",
            "targetColumn": "without_symbols"
        }
    }
}
```

REMOVE_WHITESPACE

從來源資料欄中的字串或自訂字串中移除空格，並在新資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "sourceColumn": "job_desc",
            "targetColumn": "job_desc_remove_whitespace"
```

```
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "value": "This string has spaces in it",
            "targetColumn": "string_without_spaces"
        }
    }
}
```

REPEAT_STRING

以指定的次數重複來源資料欄或自訂輸入值中的字串，並在新的資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `count` – 重複字串的次數。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "REPEAT_STRING",
        "Parameters": {
            "count": 3,
```

```
            "sourceColumn": "last_name",
            "targetColumn": "last_name_repeat_string"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REPEAT_STRING",
        "Parameters": {
            "count": 80,
            "value": "*",
            "targetColumn": "80_stars"
        }
    }
}
```

RIGHT

根據字元數，會從來源資料欄或自訂字串取得字串中最右邊的字元數，並傳回新資料欄中最右邊的指定字元數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `position` – 從字串右側開始的字元位置。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "RIGHT",
        "Parameters": {
            "sourceColumn": "nationality",
            "position": "3",
            "targetColumn": "nationality_right"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "RIGHT",
        "Parameters": {
            "value": "United States of America",
            "position": "7",
            "targetColumn": "usa_right"
        }
    }
}
```

RIGHT_FIND

從右到左搜尋，從來源資料欄或自訂值尋找符合指定字串的字串，並在新資料欄傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `pattern` – 要搜尋的規則表達式。
- `position` – 從字串右端開始的字元位置。
- `ignoreCase` – 如果為 `true`，請忽略字母之間的大小寫差異 （大小寫之間）。若要強制執行嚴格的比對，請`false`改用 。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
```

```
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

如果指定的最左邊字元數或自訂字串符合模式，`true`則會在新資料欄中傳回 。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `pattern` – 必須與字串開頭相符的規則運算式。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STARTS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
        }
    }
}
```

STRING_GREATER_THAN

建立填入下列其中一項的新資料欄：

- `True` 如果資料欄（或值）中的一個字串大於不同資料欄（或值）中的另一個字串。
- `False` 如果沒有相符項目。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - 兩者都是 `sourceColumn`*N*。
> - 其中一個 `sourceColumn`*N*和其中一個 `value`*N*。
> - 兩者都是 `value`*N*。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
        }
    }
}
```

STRING_GREATER_THAN_EQUAL

建立填入下列其中一項的新資料欄：

- `True` 如果資料欄（或值）中的一個字串大於或等於不同資料欄（或值）中的另一個字串。
- `False` 如果沒有相符項目。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - 兩者都是 `sourceColumn`*N*。
> - 其中一個 `sourceColumn`*N*和其中一個 `value`*N*。
> - 兩者都是 `value`*N*。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "nationality",
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

STRING_LESS_THAN

建立填入下列其中一項的新資料欄：

- `True` 如果資料欄（或值）中的一個字串小於不同資料欄（或值）中的另一個字串。
- `False` 如果沒有相符項目。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - 兩者都是 `sourceColumn`*N*。
> - 其中一個 `sourceColumn`*N*和其中一個 `value`*N*。
> - 兩者都是 `value`*N*。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

STRING_LESS_THAN_EQUAL

建立填入下列其中一項的新資料欄：

- `True` 如果資料欄（或值）中的一個字串小於或等於不同資料欄（或值）中的另一個字串。
- `False` 如果沒有相符項目。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - 兩者都是 `sourceColumn`*N*。
> - 其中一個 `sourceColumn`*N*和其中一個 `value`*N*。
> - 兩者都是 `value`*N*。

Example範例

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

根據使用者定義的開始和結束索引值，在新資料欄中傳回來源資料欄中部分或全部的指定字串。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `startPosition` – 從字串左端開始的字元位置。
- `endPosition` – 從字串左端到結尾的字元位置。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
}
```

TRIM

從來源資料欄或自訂字串中的字串中移除前置和後置空格，並在新資料欄中傳回結果。不會移除單字之間的空格。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
    }
}
```

UNICODE

在新資料欄中傳回來源資料欄中字串的第一個字元或自訂字串的 Unicode 索引值。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。

- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
        }
    }
}
```

UPPER

將來源資料欄或自訂字串中的所有字母字元轉換為大寫，並在新的資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

> **Note**
>
> 您可以指定 sourceColumn 或 value，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
        }
    }
}
```

日期和時間函數

接下來，尋找使用配方動作的日期和時間函數參考主題。

主題

- CONVERT_TIMEZONE
- DATE
- DATE_ADD
- DATE_DIFF
- DATE_FORMAT

- DATE_TIME
- DAY
- HOUR
- 毫秒
- MINUTE
- MONTH
- MONTH_NAME
- NOW
- 季度
- SECOND
- TIME
- 今天
- UNIX_TIME
- UNIX_TIME_FORMAT
- WEEK_DAY
- WEEK_NUMBER
- YEAR

CONVERT_TIMEZONE

根據指定的時區，將來源資料欄的時間值轉換為新的資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。來源資料欄可以是類型 `string`、 `date`或 `timestamp`。
- `fromTimeZone` – 來源值時區。如果未指定任何項目，則預設時區為 UTC。
- `toTimeZone` – 要轉換為的時區。如果未指定任何項目，則預設時區為 UTC。
- `targetColumn` – 新建立資料欄的名稱。
- `dateTimeFormat` - 選用。日期的格式字串。如果未指定格式，則會使用預設格式：`yyyy-mm-dd HH:MM:SS`。

Example範例

```
{
    "RecipeAction": {
        "Operation": "CONVERT_TIMEZONE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "fromTimeZone": "UTC+08:00",
            "toTimeZone": "UTC+08:00",
            "targetColumn": "DATETIME Column CONVERT_TIMEZONE",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS"
        }
    }
}
```

DATE

從來源資料欄或提供的值，建立包含日期值的新資料欄。

Parameters

- dateTimeFormat - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定此字串，預設格式為 yyyy-mm-dd HH:MM:SS。
- dateTimeParameters – JSON 編碼字串，代表日期和時間的元件：
  - year
  - value
  - month
  - day
  - hour
  - second

  每個元件都必須指定下列其中一項：
  - sourceColumn – 現有資料欄的名稱。
  - value – 要評估的字元字串。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
```

```
    "RecipeAction": {
        "Operation": "DATE",
        "Parameters": {
            "dateTimeFormat": "mm/dd/yy",
            "dateTimeParameters": "{\"year\":{\"value\":\"2019\"},\"month\":{\"value\":
\"12\"},\"day\":{\"value\":\"31\"},\"hour\":{},\"minute\":{},\"second\":{}}",
            "targetColumn": "DATE Column 1"
        }
    }
}
```

DATE_ADD

從來源資料欄或值將年、月或日新增至日期，並建立新的資料欄，其中包含結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `units` – 用於調整日期的度量單位。有效值為 `MONTHS`、`YEARS`、`MILLISECONDS`、`QUARTERS`、`HOURS`、`MICROSECONDS`、`WEEKSSECONDS`、、`DAYS`和 `MINUTES`。
- `dateAddValue` – `units`要新增至日期的 數目。
- `dateTimeFormat` - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定，則預設格式為 `yyyy-mm-dd HH:MM:SS`。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
```

```
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
}
```

DATE_DIFF

建立新的資料欄，其中包含兩個日期之間的差異。

Parameters

- `sourceColumn1` – 現有資料欄的名稱。
- `sourceColumn2` – 現有資料欄的名稱。
- `value1` – 要評估的字元字串。
- `value2` – 要評估的字元字串。
- `units` – 描述日期之間差異的度量單位。有效值為 `MONTHS`、`YEARS`、`MILLISECONDS`、`QUARTERS`、`HOURS`、`MICROSECONDS`、`WEEKSSECONDS`、、`DAYS`和 `MINUTES`。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您只能指定下列其中一個組合：
>
> - `sourceColumn1` 和 兩者`sourceColumn2`。
> - `sourceColumn1` 或 之一`sourceColumn2`，以及 `value1`或 之一`value2`。
> - `value1` 和 兩者`value2`。

Example範例

```
{
    "RecipeAction": {
```

```
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
        }
    }
}
```

DATE_FORMAT

從代表日期的字串建立包含特定格式日期的新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字串。
- `dateTimeFormat` - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定，則預設格式為 `yyyy-mm-dd HH:MM:SS`。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DATE_FORMAT",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "dateTimeFormat": "month*dd*yyyy",
            "targetColumn": "DATE Column 1_DATEFORMAT"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "DATE_FORMAT",
        "Parameters": {
            "value": "22:10:47",
            "dateTimeFormat": "HH:MM:SS",
            "targetColumn": "formatted_date_value"
        }
    }
}
```

DATE_TIME

從來源資料欄或提供的值，建立包含日期和時間值的新資料欄。

Parameters

- `dateTimeFormat` - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定此字串，預設格式為 `yyyy-mm-dd HH:MM:SS`。
- `dateTimeParameters` – JSON 編碼字串，代表日期和時間的元件：
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  每個元件都必須指定下列其中一項：
  - `sourceColumn` – 現有資料欄的名稱。
  - `value` – 要評估的字元字串。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DATE_TIME",
        "Parameters": {
```

```
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "dateTimeParameters": "{\"year\":{\"value\":\"2010\"},\"month\":{\"value\":
\"5\"},\"day\":{\"value\":\"21\"},\"hour\":{\"value\":\"13\"},\"minute\":{\"value\":
\"34\"},\"second\":{\"value\":\"25\"}}",
            "targetColumn": "DATETIME Column 1"
        }
    }
}
```

DAY

從代表日期的字串建立新的資料欄，其中包含月份的日期。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

HOUR

從代表日期的字串建立包含小時值的新資料欄。

Parameters

- sourceColumn – 現有資料欄的名稱。
- value – 要評估的字元字串。
- targetColumn – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 sourceColumn 或 value，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
}
```

毫秒

建立新的資料欄，其中包含來源資料欄或輸入值的毫秒值。

Parameters

- sourceColumn – 現有資料欄的名稱。來源資料欄可以是類型 string、 date或 timestamp。
- value – 要評估的字元字串。
- targetColumn – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 sourceColumn 或 value，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
}
```

MINUTE

從代表日期的字串建立包含分鐘值的新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
```

```
}
```

MONTH

從代表日期的字串建立新的資料欄，其中包含月份的數字。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

MONTH_NAME

從代表日期的字串建立新的資料欄，其中包含月份名稱。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。

- targetColumn – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 sourceColumn 或 value，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

NOW

建立包含目前日期和時間的新資料欄，格式為 yyyy-mm-dd HH:MM:SS。

Parameters

- timeZone – 時區的名稱。如果未指定時區，則預設值為國際標準時間 (UTC)。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
```

```
}
```

季度

從代表日期的字串建立新的資料欄，其中包含以日期為基礎的季度。

> **Note**
>
> 新資料欄中的季度指定為 1、2、3 或 4。
>
> - 1 為 1 月、2 月和 3 月。
> - 2 是 4 月、5 月和 6 月。
> - 3 是 7 月、8 月和 9 月。
> - 4 是 10 月、11 月和 12 月。

Parameters

- `sourceColumn` – 現有資料欄的名稱。來源資料欄可以是類型 `string`、 `date`或 `timestamp`。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
```

```
}
```

SECOND

從代表日期的字串建立包含第二個值的新資料欄。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

從提供的來源資料欄或值建立包含時間值的新資料欄。

Parameters

- `dateTimeFormat` - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定此字串，預設格式為 `yyyy-mm-dd HH:MM:SS`。

- dateTimeParameters – JSON 編碼字串，代表日期和時間的元件：
  - year
  - value
  - month
  - day
  - hour
  - second

  每個元件都必須指定下列其中一項：
  - sourceColumn – 現有資料欄的名稱。
  - value – 要評估的字元字串。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TIME",
        "Parameters": {
            "dateTimeFormat": "HH:MM:SS",
            "dateTimeParameters": "{\"year\":{},\"month\":{},\"day\":{},\"hour\":
{\"sourceColumn\":\"rand_hour\"},\"minute\":{\"sourceColumn\":\"rand_minute\"},\"second
\":{\"sourceColumn\":\"rand_second\"}}",
            "targetColumn": "TIME Column 1"
        }
    }
}
```

今天

建立包含目前日期的新資料欄，格式為 yyyy-mm-dd。

Parameters

- timeZone – 時區的名稱。如果未指定時區，則預設值為國際標準時間 (UTC)。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
        }
    }
}
```

UNIX_TIME

根據來源資料欄或輸入值，建立新的資料欄，其中包含代表 epoch 時間 (Unix 時間） 的數字，也就是自 1970 年 1 月 1 日起的秒數。如果可以推斷時區，輸出會位於該時區。否則，輸出會以國際標準時間 (UTC) 顯示。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
```

```
        }
    }
}
```

UNIX_TIME_FORMAT

將來源資料欄或輸入值的 Unix 時間轉換為指定的數值日期格式，並在新的資料欄中傳回結果。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 代表 Unix epoch 時間戳記的整數。
- `dateTimeFormat` - 選用。日期的格式字串，因為它會出現在新的欄中。如果未指定，則預設格式為 `yyyy-mm-dd HH:MM:SS`。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

WEEK_DAY

從代表日期的字串建立新的資料欄，其中包含星期幾。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

WEEK_NUMBER

從代表日期的字串建立新的資料欄，其中包含週數 （ 從 1 到 52)。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

YEAR

從代表日期的字串建立新的資料欄，其中包含年份。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 新建立資料欄的名稱。

> **Note**
>
> 您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
```

```
}
```

範圍函數

接下來，尋找使用配方動作之視窗函數的參考主題。

主題

- FILL
- NEXT
- 上一個
- ROLLING_AVERAGE
- ROLLING_COUNT_A
- ROLLING_KTH_LARGEST
- ROLLING_KTH_LARGEST_UNIQUE
- ROLLING_MAX
- ROLLING_MIN
- ROLLING_MODE
- ROLLING_STANDARD_DEVIATION
- ROLLING_SUM
- ROLLING_VARIANCE
- ROW_NUMBER
- SESSION

FILL

根據指定的來源資料欄傳回新的資料欄。對於來源欄中的任何遺失或 null 值， 會從有問題的來源值前後的資料列視窗中FILL選擇最新的非空白值。所選值接著會放置在新的資料欄中。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。

- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
}
```

NEXT

傳回新資料欄，其中每個值代表來源資料欄中稍後 n 個資料列的值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- numRows – 代表來源資料欄中稍早 n 個資料列的值。例如，如果 numRows是 3，則 NEXT 會使用第三個下一個sourceColumn值做為新targetColumn值。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
```

```
}
```

上一個

傳回新資料欄，其中每個值代表來源資料欄中稍早 n 個資料列的值。

Parameters

- sourceColumn – 現有資料欄的名稱。
- numRows – 代表來源資料欄中稍早 n 個資料列的值。例如，如果 numRows是 3，則 PREV會使用第三個先前sourceColumn值做為新targetColumn值。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

ROLLING_AVERAGE

在新資料欄中傳回從 之前指定資料列數的值滾動平均值，到指定資料欄中目前資料列之後的指定資料列數。

Parameters

- sourceColumn – 現有資料欄的名稱。
- numRowsBefore – 目前來源資料列之前的列數，代表視窗的開頭。
- numRowsAfter – 目前來源資料列後面的列數，代表視窗的結尾。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
    }
}
```

ROLLING_COUNT_A

在新資料欄中傳回從指定資料列前的指定資料列數到指定資料列後指定資料列數的非 Null 值滾動計數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

在新資料欄中傳回從之前指定資料列數的滾動第 k 個最大值，到指定資料欄中目前資料列之後的指定資料列數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。
- `value` – k 的值。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
  "Action": {
    "Operation": "ROLLING_KTH_LARGEST",
    "Parameters": {
      "sourceColumn": "weight_kg",
      "numRowsBefore": "5",
      "numRowsAfter": "5",
      "value": "3"
      "targetColumn": "weight_kg_ROLLING_KTH_LARGEST"
    }
  }
}
```

ROLLING_KTH_LARGEST_UNIQUE

在新資料欄中傳回從之前指定資料列數滾動的唯一 k 最大值，到指定資料欄中目前資料列後面指定的資料列數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。

- value – k 的值。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
  "Action": {
    "Operation": "ROLLING_KTH_LARGEST_UNIQUE",
    "Parameters": {
      "sourceColumn": "games_played",
      "numRowsBefore": "3",
      "numRowsAfter": "3",
      "value": "5",
      "targetColumn": "weight_kg_ROLLING_KTH_LARGEST_UNIQUE"
    }
  }
}
```

ROLLING_MAX

在新資料欄中傳回從之前指定資料列數到指定資料欄中目前資料列後指定資料列數的值滾動最大值。

Parameters

- sourceColumn – 現有資料欄的名稱。

  numRowsBefore – 目前來源資料列之前的列數，代表視窗的開頭。
- numRowsAfter – 目前來源資料列後面的列數，代表視窗的結尾。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_MAX",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
```

```
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

ROLLING_MIN

在新資料欄中傳回從之前指定資料列數到指定資料欄中目前資料列後指定資料列數的滾動最小值。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

  `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

ROLLING_MODE

在新資料欄中傳回滾動模式（最常見的值），從 之前的指定資料列數到指定資料欄中目前資料列後面的指定資料列數。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

- numRowsBefore – 目前來源資料列之前的列數，代表視窗的開頭。
- numRowsAfter – 目前來源資料列後面的列數，代表視窗的結尾。
- modeType – 要套用至視窗的模態函數。有效值為 NONE、MINIMUM、MAXIMUM、AVERAGE。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

ROLLING_STANDARD_DEVIATION

在新資料欄中傳回從指定資料列數目的 值滾動標準差，到指定資料欄中目前資料列後面的指定資料列數目。

Parameters

- sourceColumn – 現有資料欄的名稱。
- numRowsBefore – 目前來源資料列之前的列數，代表視窗的開頭。
- numRowsAfter – 目前來源資料列後面的列數，代表視窗的結尾。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
```

```
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

ROLLING_SUM

在新資料欄中，傳回從指定資料列前的指定資料列數到指定資料列後指定資料列數的值滾動總和。

Parameters

- `sourceColumn` – 現有資料欄的名稱。

  `numRowsBefore` – 目前來源資料列之前的列數，代表視窗的開頭。
- `numRowsAfter` – 目前來源資料列後面的列數，代表視窗的結尾。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

ROLLING_VARIANCE

在新資料欄中傳回從 之前指定資料列數到指定資料欄中目前資料列後指定資料列數的值滾動差異。

Parameters

- sourceColumn – 現有資料欄的名稱。
- numRowsBefore – 目前來源資料列之前的列數，代表視窗的開頭。
- numRowsAfter – 目前來源資料列後面的列數，代表視窗的結尾。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

在新資料欄中，根據「群組依據」和「排序依據」陳述式的資料欄名稱所建立的時段，傳回工作階段識別符。

Parameters

- groupByColumns – 描述「group by」資料欄的 JSON 編碼字串。
- orderByColumns – 描述「排序依據」欄的 JSON 編碼字串。
- targetColumn – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
```

```
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESSION

在新資料欄中，根據「群組依據」和「排序依據」陳述式的資料欄名稱所建立的時段，傳回工作階段識別符。

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `units` – 描述工作階段長度的度量單位。有效值為 `MONTHS`、`YEARS`、`MILLISECONDS`、`QUARTERS`、`HOURS`、`MICROSECONDS`、`WEEKSSECONDS`、、`DAYS`和 `MINUTES`。
- `value` – `units`用來定義時段的 數目。
- `groupByColumns` – 描述「group by」資料欄的 JSON 編碼字串。
- `orderByColumns` – 描述「排序依據」欄的 JSON 編碼字串。
- `targetColumn` – 新建立資料欄的名稱。

Example範例

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
```

```
}
```

Web 函數

接下來，尋找使用配方動作之 Web 函數的參考主題。

主題

- IP_TO_INT
- INT_TO_IP
- URL_PARAMS

IP_TO_INT

將來源資料欄或其他值的網際網路通訊協定第 4 版 (IPv4) 值轉換為目標資料欄中對應的整數值，並在新的資料欄中傳回結果。此函數僅適用於 IPv4。

例如，請考慮下列 IP 地址。

```
192.168.1.1
```

如果您使用此值做為 的輸入`IP_TO_INT`，輸出值如下所示。

```
3232235777
```

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
```

```
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_TO_IP

將來源資料欄或其他值的整數值轉換為目標資料欄中對應的 IPv4 值，並在新的資料欄中傳回結果。此函數僅適用於 IPv4。

例如，請考慮下列整數。

```
167772410
```

如果您使用此值做為 的輸入INT_TO_IP，輸出值如下所示。

```
10.0.0.250
```

Parameters

- sourceColumn – 現有資料欄的名稱。
- value – 要評估的字元字串。
- targetColumn – 要建立的新資料欄的名稱。

您可以指定 sourceColumn 或 value，但不能同時指定兩者。

Example範例

```
[  {
    "RecipeAction": {
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
```

```
  }
]
```

URL_PARAMS

從 URL 字串擷取查詢參數，將其格式化為 JSON 物件，並在新資料欄中傳回結果。

例如，請考慮下列 URL。

```
https://example.com/?firstParam=answer&secondParam=42
```

如果您使用此值做為 的輸入URL_PARAMS，輸出值如下所示。

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parameters

- `sourceColumn` – 現有資料欄的名稱。
- `value` – 要評估的字元字串。
- `targetColumn` – 要建立的新資料欄的名稱。

您可以指定 `sourceColumn` 或 `value`，但不能同時指定兩者。

Example範例

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
    }
}
```

其他 函數

接下來，尋找其他使用配方動作之函數的參考主題。

主題

COALESCE

在新資料欄中傳回在資料欄陣列中找到的第一個非空值。函數中列出的資料欄順序會決定搜尋資料的順序。

Parameters

- `sourceColumns` – 代表現有資料欄清單的 JSON 編碼字串。
- `targetColumn` – 要建立的新資料欄的名稱。

Example範例

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

擷取先前提交動作的結果。僅適用於互動式體驗。

Parameters

- `actionId` – 在原始 SendProjectSessionAction 回應中傳回的 ActionId。

Example範例

```
{
    "RecipeAction": {
        "Operation": "GET_ACTION_RESULT",
        "Parameters": {
            "actionId": "7",
        }
    }
}
```

GET_STEP_DATAFRAME

從專案配方中的步驟擷取資料框架。僅適用於互動式體驗。與 ViewFrame 參數搭配使用，以跨大型資料框架進行分頁。

Parameters

- `stepIndex` – 專案配方中要擷取資料框架之步驟的索引。

Example範例

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

的配額AWS Glue DataBrew

您可以在 AWS Service Quotas 主控台中檢視 DataBrew 服務配額。您也可以針對任何可調整的配額請求增加配額。

AWS Glue DataBrew開發人員指南的文件歷史記錄

目前的 API 版本：databrew-2017-07-25

下表說明此版本 的文件AWS Glue DataBrew。如果您想要在AWS Glue DataBrew開發人員指南更新時收到通知，您可以訂閱 RSS 摘要。

變更	描述	日期
glue:GetCustomEntityType 已新增至AWS受管政策	執行已啟用 PII 身分的AWS Glue DataBrew設定檔任務需要此許可。如需詳細資訊，請參閱 AWS Glue DataBrewAWS受管政策的更新。	2024 年 3 月 20 日
在 CRYPTOGRAPHIC_HASH 轉換中支援多個雜湊演算法	您現在可以在資料欄中雜湊值時指定雜湊演算法。如需詳細資訊，請參閱 CRYPTOGRAPHIC_HASH。	2023 年 8 月 11 日
glue:BatchGetCustomEntityTypes 已新增至AWS受管政策	執行已啟用 PII 身分的AWS Glue DataBrew設定檔任務需要此許可。如需詳細資訊，請參閱 AWS Glue DataBrewAWS受管政策的更新。	2022 年 5 月 9 日
支援 Apache ORC 檔案格式	DataBrew 現在支援 Apache ORC 做為 DataBrew 資料來源和輸出的檔案格式。如需詳細資訊，請參閱資料來源支援的檔案類型。	2022 年 3 月 31 日
支援跨帳戶AWS Glue Data Catalog Amazon S3 存取	AWS 帳戶如果AWS Glue已在主控台中建立適當的資源政策，您現在可以從其他 存取 AWS Glue Data Catalog S3 資料表。建立政策後，您可以在建立 DataBrew 資料集時選取	2022 年 3 月 11 日

	相關的 Data Catalog S3 資料表做為輸入來源。如需詳細資訊，請參閱資料來源和輸出支援的連線。	
支援原生主控台與 Amazon AppFlow 整合	DataBrew 現在已與 Amazon AppFlow 進行原生主控台整合。此整合表示您可以從 Salesforce、Zendesk、Slack、ServiceNow 和其他 software-as-a-service(SaaS) 應用程式連線至資料。您也可以從 Amazon S3 和 Amazon Redshift AWS 服務等 連線到資料。如需詳細資訊，請參閱資料來源和輸出支援的連線。	2021 年 11 月 18 日
支援資料品質規則	DataBrew 現在支援建立資料品質規則，這是可自訂的驗證檢查，可定義特定資料的業務需求。如需詳細資訊，請參閱在 中驗證資料品質AWS Glue DataBrew。	2021 年 11 月 18 日
支援自訂 SQL 陳述式	DataBrew 現在支援從 Amazon Redshift 和 Snowflake 擷取資料的自訂 SQL 陳述式。此支援表示您可以使用專用查詢來選取和限制從大型資料表傳回的資料。如需詳細資訊，請參閱資料來源和輸出支援的連線。	2021 年 11 月 18 日

支援 PII 偵測	DataBrew 現在支援偵測個人身分識別資訊 (PII)。這可讓您選擇在資料準備期間遮罩 PII。如需詳細資訊，請參閱識別和處理個人身分識別資訊 (PII)。	2021 年 11 月 18 日
支援其他AWS區域	DataBrew 現在支援其他AWS區域。如需支援區域的清單，請參閱AWS Glue DataBrew端點和配額。	2021 年 10 月 5 日
支援將資料寫入 Lake Formation 型 Amazon S3 資料表	DataBrew 現在支援根據 將資料寫入AWS Glue Data Catalog S3 資料表AWS Lake Formation。DataBrew 現在也支援將資料寫入 Tableau Hyper 格式。如需詳細資訊，請參閱建立和使用AWS Glue DataBrew配方任務。	2021 年 8 月 13 日
支援將資料寫入 JDBC 目的地	DataBrew 現在支援將資料直接寫入 JDBC 支援的資料庫和資料倉儲。其中包括 Amazon Redshift、Snowflake、Microsoft SQL Server、My SQL、Oracle Database 和 PostgreSQL。如需詳細資訊，請參閱建立和使用AWS Glue DataBrew配方任務。	2021 年 7 月 23 日
支援指定為設定檔任務產生哪些資料品質統計資料	DataBrew 現在支援指定為設定檔任務中的資料集自動產生哪些資料品質統計資料。如需詳細資訊，請參閱建立和使用AWS Glue DataBrew配方任務。	2021 年 7 月 23 日

支援將資料集寫入AWS Glue Data Catalog	DataBrew 現在支援將資料集直接寫入AWS Glue Data Catalog。您可以選擇將從執行資料準備配方的任務建立的資料集存放在 Data Catalog 的 Amazon S3、Amazon Redshift 和 Amazon RDS 資料表中。支援的 RDS 資料表包括 Amazon Aurora、RDS for Oracle、RDS for Microsoft SQL Server、RDS for MySQL 和 RDS for PostgreSQL。	2021 年 6 月 30 日
支援識別進階資料類型	DataBrew 現在支援自動識別和標記資料欄的進階資料類型，這可讓您更輕鬆地標準化包含特定資料類型的資料欄。這些類型的資料包括社會安全號碼、電子郵件地址、電話號碼、性別、信用卡、URL、IP 地址、日期和時間、貨幣、郵遞區號、國家、區域、州和城市。	2021 年 6 月 30 日
支援使用 Amazon AppFlow 從 SAAS 應用程式傳輸資料	DataBrew 現在支援使用 Amazon AppFlow，將資料從第三方software-as-a-service(SaaS) 應用程式傳輸至 Amazon S3，例如 Salesforce、Zendesk、Slack 和 ServiceNow。如需詳細資訊，請參閱資料來源和輸出支援的連線。	2021 年 4 月 29 日

支援使用來自 JDBC 資料庫的輸入建立 DataBrew 資料集	DataBrew 現在支援從 JDBC 支援的資料庫和資料倉儲中的資料建立資料集，包括 Amazon Redshift、Snowflake、Microsoft SQL Server、MySQL、Oracle Database 和 PostgreSQL。如需詳細資訊，請參閱資料來源和輸出支援的連線。	2021 年 4 月 2 日
支援其他AWS 區域	DataBrew 現在支援其他AWS 區域。如需支援區域的清單，請參閱AWS Glue DataBrew端點和配額。	2021 年 1 月 28 日
處理重複項目的新轉換	處理重複的四個新轉換已新增至 DataBrew 主控台和 API。如需詳細資訊，請參閱資料品質配方步驟中的 DELETE_DUPLICATE_ROWS、FLAG_DUPLICATE_ROWS、FLAG_DUPLICATES_IN_COLUMN 和 REMOVE_DUPLICATES。	2021 年 1 月 28 日
其他 CSV 分隔符號	DataBrew 現在除了用於建立 DataBrew 資料集的逗號分隔值 (CSV) 檔案中的逗號之外，還支援其他分隔符號。如需詳細資訊，請參閱建立和使用AWS Glue DataBrew資料集。	2021 年 1 月 28 日
JupyterLab 的 DataBrew 延伸模組	現在您可以在 JupyterLab 中使用AWS Glue DataBrew作為延伸。如需詳細資訊，請參閱在 JupyterLab 中使用 DataBrew 作為延伸模組。	2020 年 11 月 20 日

新的資料準備工具：AWS Glue DataBrew	這是《AWS Glue DataBrew開發人員指南》的第一版。	2020 年 11 月 11 日

AWS詞彙表

如需最新的AWS術語，請參閱 AWS 詞彙表參考中的AWS詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。