aws

Unable to locate subtitle

AWS Glue DataBrew Guia do Desenvolvedor

AWS Glue DataBrew Guia do Desenvolvedor: ***Unable to locate subtitle***

Table of Contents

O que éAWS Glue DataBrew?

AWS Glue DataBrewé uma ferramenta visual de preparação de dados que permite aos usuários limpar e normalizar dados sem escrever nenhum código. O uso DataBrew ajuda a reduzir o tempo necessário para preparar dados para análise e aprendizado de máquina (ML) em até 80%, em comparação com a preparação de dados desenvolvida de forma personalizada. Você pode escolher entre mais de 250 transformações prontas para automatizar tarefas de preparação de dados, como filtrar anomalias, converter dados em formatos padrão e corrigir valores inválidos.

Usando DataBrew, analistas de negócios, cientistas de dados e engenheiros de dados podem colaborar mais facilmente para obter insights de dados brutos. Por ser DataBrew sem servidor, não importa qual seja seu nível técnico, você pode explorar e transformar terabytes de dados brutos sem precisar criar clusters ou gerenciar qualquer infraestrutura.

Com a DataBrew interface intuitiva, você pode descobrir, visualizar, limpar e transformar dados brutos de forma interativa. DataBrew faz sugestões inteligentes para ajudá-lo a identificar problemas de qualidade de dados que podem ser difíceis de encontrar e demorados de corrigir. DataBrew Ao preparar seus dados, você pode usar seu tempo para agir sobre os resultados e iterar mais rapidamente. Você pode salvar a transformação como etapas em uma receita, que pode ser atualizada ou reutilizada posteriormente com outros conjuntos de dados e implantá-la continuamente.

A imagem a seguir mostra como DataBrew funciona em alto nível.

Para usar DataBrew, você cria um projeto e se conecta aos seus dados. No espaço de trabalho do projeto, você vê seus dados exibidos em uma interface visual em forma de grade. Aqui, você pode explorar os dados e ver distribuições de valores e gráficos para entender seu perfil.

Para preparar os dados, você pode escolher entre mais de 250 transformações de apontar e clicar. Isso inclui remover nulos, substituir valores ausentes, corrigir inconsistências de esquema, criar colunas com base em funções e muito mais. Você também pode usar transformações para aplicar técnicas de processamento de linguagem natural (PNL) para dividir frases em frases. As visualizações imediatas mostram uma parte dos seus dados antes e depois da transformação, para que você possa modificar sua receita antes de aplicá-la a todo o conjunto de dados.

Depois DataBrew de executar sua receita em seu conjunto de dados, a saída é armazenada no Amazon Simple Storage Service (Amazon S3). Depois que seu conjunto de dados limpo e preparado estiver no Amazon S3, outro dos seus sistemas de armazenamento ou gerenciamento de dados poderá ingeri-lo.

Conceitos e termos básicos em AWS Glue DataBrew

A seguir, você pode encontrar uma visão geral dos principais conceitos e terminologia em AWS Glue DataBrew. Depois de ler esta seção, consulteConceitos básicos de AWS Glue DataBrew, que orienta você no processo de criação de projetos, conexão de conjuntos de dados e execução de trabalhos.

Tópicos

- Projeto
- Conjunto de dados
- Fórmula
- Trabalho
- Linhagem de dados
- Perfil de dados

Projeto

O espaço de trabalho interativo de preparação de dados em DataBrew é chamado de projeto. Usando um projeto de dados, você gerencia uma coleção de itens relacionados: dados, transformações e processos agendados. Como parte da criação de um projeto, você escolhe ou cria um conjunto de dados para trabalhar. Em seguida, você cria uma receita, que é um conjunto de

instruções ou etapas que você DataBrew deseja seguir. Essas ações transformam seus dados brutos em um formulário pronto para ser consumido pelo seu pipeline de dados.

Conjunto de dados

Conjunto de dados significa simplesmente um conjunto de dados — linhas ou registros divididos em colunas ou campos. Ao criar um DataBrew projeto, você se conecta ou carrega dados que deseja transformar ou preparar. DataBrew pode trabalhar com dados de qualquer fonte, importados de arquivos formatados, e se conecta diretamente a uma lista crescente de armazenamentos de dados.

Pois DataBrew, um conjunto de dados é uma conexão somente para leitura com seus dados. DataBrew coleta um conjunto de metadados descritivos para se referir aos dados. Nenhum dado real pode ser alterado ou armazenado pelo DataBrew. Para simplificar, usamos o conjunto de dados para nos referirmos tanto ao conjunto de dados real quanto aos usos dos metadados DataBrew .

Fórmula

Em DataBrew, uma receita é um conjunto de instruções ou etapas para dados com os quais você DataBrew deseja agir. Uma receita pode conter várias etapas e cada etapa pode conter muitas ações. Você usa as ferramentas de transformação na barra de ferramentas para configurar todas as alterações que deseja fazer nos seus dados. Posteriormente, quando estiver pronto para ver o produto final de sua receita, você atribui esse trabalho DataBrew e o agenda. DataBrew armazena as instruções sobre a transformação de dados, mas não armazena nenhum dos seus dados reais. Você pode baixar e reutilizar receitas em outros projetos. Você também pode publicar várias versões de uma receita.

Trabalho

DataBrew assume a tarefa de transformar seus dados executando as instruções que você configurou ao criar uma receita. O processo de execução dessas instruções é chamado de trabalho. Um trabalho pode colocar suas receitas de dados em ação de acordo com um cronograma predefinido. Mas você não está confinado a um cronograma. Você também pode executar trabalhos sob demanda. Se você quiser criar o perfil de alguns dados, não precisará de uma receita. Nesse caso, basta configurar um trabalho de perfil para criar um perfil de dados.

Linhagem de dados

DataBrew rastreia seus dados em uma interface visual para determinar sua origem, chamada de linhagem de dados. Essa exibição mostra como os dados fluem por diferentes entidades de onde

vieram originalmente. Você pode ver sua origem, outras entidades pelas quais ela foi influenciada, o que aconteceu com ela ao longo do tempo e onde ela foi armazenada.

Perfil de dados

Quando você cria o perfil de seus dados, DataBrew cria um relatório chamado perfil de dados. Esse resumo fala sobre a forma existente dos seus dados, incluindo o contexto do conteúdo, a estrutura dos dados e suas relações. Você pode criar um perfil de dados para qualquer conjunto de dados executando um trabalho de perfil de dados.

Integrações de produtos e serviços

Use esta seção para saber com quais produtos e serviços se integram DataBrew.

DataBrew trabalha com os seguintes AWS serviços de rede, gerenciamento e governança:

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

DataBrew funciona com os seguintes AWS data lakes e armazenamentos de dados:

- AWS Lake Formation
- Amazon S3

DataBrew suporta os seguintes formatos de arquivo e extensões para upload de dados.

Formato	Extensão do arquivo (opcional)	Extensões para arquivos compactados (obrigatório)

Comma-separated valores	`.csv`	`.gz`
		`.snappy`
		`.lz4`

Formato	Extensão do arquivo (opcional)	Extensões para arquivos compactados (obrigatório)
		`.bz2` `.deflate`
Pasta de trabalho do Microsoft Excel	`.xlsx`	Sem suporte para compressão

JSON (documento JSON e linhas JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

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

DataBrew grava arquivos de saída no Amazon S3 e oferece suporte aos seguintes formatos e extensões de arquivo.

Formato	Extensão de arquivo (não compactada)	Extensões de arquivo (compactadas)
Comma-separated valores	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`

Formato	Extensão de arquivo (não compactada)	Extensões de arquivo (compactadas)
Tab-separated valores	`.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	Não compatível	`.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 (somente no formato de linhas JSON)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	Não compatível	Não aplicável

Configurar AWS Glue DataBrew

Antes de começar AWS Glue DataBrew, você precisa configurar algumas permissões, um usuário e uma função. Comece executando as seguintes etapas:

1. Inscrever-se AWS em uma conta conforme necessário e criando políticas AWS Identity and Access Management(IAM) para permitir que os usuários executem DataBrew:
   - Inscrever-se em uma nova AWS conta e adicionar um usuário. Para obter mais informações, consulte Configurando um novo AWS account.
   - Adicionar uma política do IAM para um usuário do console. Um usuário com essas permissões pode acessar DataBrew no Console de gerenciamento da AWS.
   - Adicionar permissões para recursos de dados para uma função do IAM. Uma função do IAM com essas permissões pode acessar dados em nome do usuário.

   Você precisa ser administrador do IAM para criar usuários, funções e políticas.
2. Adicionar usuários ou grupos para DataBrew. Um usuário ou grupo com as permissões corretas anexadas pode acessar DataBrew no console.
3. Adicionar uma função com permissões para acessar dados DataBrew. Uma função com as permissões corretas pode acessar dados em nome do usuário.

Configurando um novo AWS account

Se você não tiver uma AWS conta, cadastre-se em uma AWS conta e crie um usuário administrador do IAM.

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra o https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática

recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

Para criar um usuário administrador, selecione uma das opções a seguir.

Selecionar uma forma de gerenciar o administrador	Para	Por	Você também pode
Centro de Identidade do IAM (Recomendado)	Usar credenciais de curto prazo para acessar a AWS. Isso está de acordo com as práticas recomendadas de segurança. Para obter informações sobre as práticas recomendadas, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.	Seguindo as instruções em Conceitos básicos no Guia do usuário do Centro de Identidade do AWS IAM.	Configure o acesso programático configurando o AWS CLI para uso Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário.
No IAM (Não recomendado)	Usar credenciais de longo prazo para acessar a AWS.	Seguindo as instruções em Criar um acesso de emergência para um usuário do IAM no Guia do usuário do IAM.	Configurar o acesso programático, com base em Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do IAM:

- O que é o IAM?
- Como se configurar com o IAM
- Criação de um usuário e grupo de administração (console)

Configurando o AWS CLI

Se você planeja usar JupyterLab a DataBrew API, certifique-se de instalar o AWS Command Line Interface(AWS CLI). Você não precisa dele para usar o DataBrew console ou realizar as etapas dos exercícios de introdução.

Para configurar o AWS CLI

1. Faça o download e configure o AWS CLI usando as etapas a seguir:
   - Instalar a AWS CLI
   - Noções básicas de configuração
2. Verifique a configuração digitando o seguinte DataBrew comando no prompt de comando.

   ```
   aws databrew help
   ```

   Se essa instrução retornar o erro "`aws: error: argument command: Invalid choice`" seguido por uma longa lista de serviços, desinstale o AWS CLI e reinstale. Essa ação não substitui sua configuração existente.

AWS CLI os comandos usam a AWS região padrão da sua configuração, a menos que você a defina com um parâmetro ou um perfil. Você pode adicionar o `--region` parâmetro a cada comando.

Se preferir, você pode adicionar um perfil nomeado em `~/.aws/config` ou `%UserProfile %/.aws/config` (no Microsoft Windows). Perfis nomeados também podem preservar outras configurações, conforme mostrado no exemplo a seguir.

```
[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
```

Configurar AWS Identity and Access Management Permissões (IAM)

Antes de começar, você precisa configurar algumas coisas no IAM. Você precisa ser administrador ou ter a ajuda de um. No entanto, se você tiver uma conta com acesso de administrador, poderá realizar essas tarefas sozinho. Você pode encontrar instruções simples para cada tarefa nesta seção.

Veja a seguir uma visão geral do que você precisa fazer:

- Como parte desse processo, você adiciona um usuário. Você não precisa adicionar um novo usuário, você pode usar um existente. Você anexa DataBrew permissões para que o usuário possa abrir o DataBrew console.
- Criar um perfil do IAM. Uma função permite determinadas ações e concede permissões quando é usada, dentro de limites. Por exemplo, ele só funciona para usuários da sua AWS conta. Você pode adicionar mais limitações posteriormente.
- Crie a política ou políticas do IAM de que você precisa. Uma política é uma lista de coisas que um usuário pode fazer. Para criar uma política, abra outra página do console e cole o texto de um arquivo baixado.

> **Note**
>
> O que fornecemos aqui são informações básicas de configuração. Recomendamos que você reserve um tempo para personalizar suas permissões para que elas atendam às suas necessidades de segurança e conformidade. Se precisar de ajuda, entre em contato com seu administrador ou com o AWS Support.

Para adicionar as permissões necessárias

1. Crie políticas do IAM para permitir que os usuários executem DataBrew fazendo o seguinte:
   - Adicione uma política personalizada do IAM para um usuário do console. Se você não precisar de uma política personalizada, poderá escolher a política AWS gerenciada em vez disso.

Basta adicioná-lo ao usuário na etapa 2. Um usuário com essas permissões pode acessar o console DataBrew de serviço.

- Adicione permissões para recursos de dados. Uma função do IAM com essas permissões pode acessar dados em nome do usuário.

Você precisa ser administrador para criar usuários, funções e políticas.

2. Adicione usuários ou grupos para DataBrew. Um usuário ou grupo com as permissões corretas anexadas pode acessar o DataBrew console.
3. Adicione uma função com permissões para acessar dados DataBrew. Uma função com as permissões corretas pode acessar dados em nome do usuário.

Configurando políticas do IAM para DataBrew

Você usa políticas do IAM para gerenciar permissões. Uma política facilita a adição de permissões relacionadas de uma só vez, em vez de uma por vez.

Recomendamos que você crie as políticas usando os mesmos nomes que fornecemos. Usamos os nomes mostrados a seguir para essas políticas em toda a documentação. Usar esses nomes também facilita se você precisar entrar em contato com o AWS Support. No entanto, você pode optar por alterar os nomes das políticas e seu conteúdo. Para obter mais informações sobre as políticas do IAM, consulte Criar uma política gerenciada pelo cliente no Guia do usuário do IAM.

Depois de criar as políticas necessárias para uso DataBrew, você as anexa aos usuários e funções. Como fazer isso será abordado posteriormente nesta seção.

Tópicos

- Adicionar uma política do IAM para um usuário do console
- Adicionar permissões para recursos de dados para uma função do IAM
- Configurando políticas do IAM para DataBrew

Adicionar uma política do IAM para um usuário do console

Configurar permissões para um usuário para o Console de gerenciamento da AWSé opcional, mas se você precisar de acesso ao console, execute esta etapa primeiro.

Para configurar permissões de acesso DataBrew no console, escolha uma das seguintes opções:

- Use a política gerenciada por AWS:`AwsGlueDataBrewFullAccessPolicy`. Se você escolher essa opção, vá para a próxima política,Adicionar permissões para recursos de dados para uma função do IAM.
- Crie a política descrita nesta seção,`AwsGlueDataBrewCustomUserPolicy`. Essa opção permite que você personalize a política com requisitos adicionais de segurança personalizados.

A política a seguir concede as permissões necessárias para executar o DataBrew console. Você fornece essas permissões usando o IAM.

Para definir a política AwsGlueDataBrewCustomUserPolicy do IAM para DataBrew (console)

1. Baixe o JSON para a política do `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, selecione Políticas.
4. Para cada política, escolha Criar política.
5. Na tela Criar política, navegue até a guia JSON.
6. Copie a declaração JSON da política que você baixou. Cole-o sobre o exemplo de declaração no editor.
7. Verifique se a política está personalizada de acordo com sua conta, requisitos de segurança e AWS recursos necessários. Se precisar fazer alterações, você pode fazê-las no editor.
8. Selecione Revisar política.

Para definir a política AwsGlueDataBrewCustomUserPolicy do IAM para DataBrew (AWS CLI)

1. Baixe o JSON para a política do `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Personalize a política conforme descrito na primeira etapa do procedimento anterior.
3. Execute o comando a seguir para criar a política.

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

Adicionar permissões para recursos de dados para uma função do IAM

Para se conectar aos dados, éAWS Glue DataBrew necessário ter uma função do IAM que possa ser transmitida em nome do usuário. A seguir, você pode descobrir como criar a política que posteriormente anexará a uma função do IAM.

A `AwsGlueDataBrewDataResourcePolicy` política concede as permissões necessárias para se conectar aos dados usando DataBrew. Para qualquer operação que acesse dados em outro AWS recurso, como acessar seus objetos no Amazon S3 DataBrew , é necessária permissão para acessar o recurso em seu nome.

Para definir a política AwsGlueDataBrewDataResourcePolicy do IAM para DataBrew (console)

1. Baixe o JSON para `AwsGlueDataBrewDataResourcePolicy`.
2. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, selecione Políticas.
4. Para cada política, escolha Criar política.
5. Na tela Criar política, navegue até a guia JSON.
6. Copie a declaração JSON da política que você baixou. Cole-o sobre o exemplo de declaração no editor.
7. Verifique se a política está personalizada de acordo com sua conta, requisitos de segurança e AWS recursos necessários. Se precisar fazer alterações, você pode fazê-las no editor.
8. Selecione Revisar política.

Para definir a política AwsGlueDataBrewDataResourcePolicy do IAM para DataBrew (AWS CLI)

1. Baixe o JSON para `AwsGlueDataBrewDataResourcePolicy`.
2. Personalize a política conforme descrito na primeira etapa do procedimento anterior.
3. Execute o comando a seguir para criar a política.

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

Configurando políticas do IAM para DataBrew

A seguir, você encontrará detalhes e exemplos sobre as políticas do IAM com as quais você pode usar DataBrew. Detalhes sobre as políticas básicas são fornecidos aqui. Além disso, há mais exemplos que não precisam ser usados DataBrew. São configurações adicionais que você pode usar em determinadas situações.

Tópicos

AwsGlueDataBrewCustomUserPolicy

A `AwsGlueDataBrewCustomUserPolicy` política concede a maioria das permissões necessárias para usar o DataBrew console. Alguns dos recursos especificados nesta política se referem aos serviços usados pelo DataBrew. Isso inclui nomes para AWS Glue Data Catalog buckets do Amazon S3, Amazon CloudWatch Logs e recursos.AWS KMSÉ semelhante à política AWS gerenciada chamada`AwsGlueDataBrewFullAccessPolicy`.

A tabela a seguir descreve as permissões concedidas por esta política.

Ação	Recurso	Descrição
`"databrew:*"`	`"*"`	Concede permissão para executar todas as operações DataBrew da API.

`"glue:GetDatabases"`
`"glue:GetPartitions"`
`"glue:GetTable"`
`"glue:GetTables"`
	`"*"`	Permite a listagem de AWS Glue bancos de dados e tabelas.

Ação	Recurso	Descrição
"glue:GetDataCatalogEncryptionSettings"		
"dataexchange:ListDataSets" "dataexchange:ListDataSetRevisions" "dataexchange:ListRevisionAssets" "dataexchange:CreateJob" "dataexchange:StartJob" "dataexchange:GetJob"	"*"	Permite a listagem de recursos do AWS Data Exchange em conjuntos de dados.
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	Permite a listagem de AWS KMS chaves a serem usadas para criptografia da saída do trabalho.
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	Permite a criptografia da saída do trabalho.
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionConfiguration"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Permite a listagem de buckets do Amazon S3 para projetos, conjuntos de dados e trabalhos. Permite enviar arquivos de saída para o S3.
"sts:GetCallerIdentity"	"*"	Obtenha informações sobre o chamador atual.

Ação	Recurso	Descrição
`"cloudtrail:LookupEvents",`	`"*"`	Permitir listar AWS CloudTrail eventos para conjuntos de dados (linhagem de dados).
`"iam:ListRoles"` `"iam:GetRole"`	`"*"`	Permite listar funções do IAM a serem usadas em projetos e trabalhos.

AwsGlueDataBrewDataResourcePolicy

A `AwsGlueDataBrewDataResourcePolicy` política concede as permissões necessárias para se conectar aos dados e configurar DataBrew.

A tabela a seguir descreve as permissões concedidas por esta política.

Ação	Recurso	Descrição
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permite que você visualize seus arquivos.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permite enviar arquivos de saída para o S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permite excluir um objeto criado por DataBrew.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*",`	Permite a listagem de buckets do Amazon S3 a

Ação	Recurso	Descrição
	`"arn:aws:s3:::bucket_name"`	partir de projetos, conjuntos de dados e trabalhos.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Permite a descriptografia de conjuntos de dados criptografados.
`"kms:GenerateDataKey"`	`"arn:aws:kms:::key/key_ids"`	Permite a criptografia da saída do trabalho.

`"ec2:DescribeVpcEndpoints"`
`"ec2:DescribeRouteTables"`
`"ec2:DeleteNetworkInterface"`
`"ec2:DescribeNetworkInterfaces"`
`"ec2:DescribeSecurityGroups"`
`"ec2:DescribeSubnets"`
`"ec2:DescribeVpcAttribute"`
`"ec2:CreateNetworkInterface"`
	`"*"`	Permite a configuração de itens de rede do Amazon EC2, como nuvens privadas virtuais (VPCs), ao executar trabalhos e projetos.

`"ec2:DeleteNetworkInterface"`	`"*"`	Permite excluir uma interface de rede em uma VPC.

Ação	Recurso	Descrição

`"ec2:CreateTags"`
 	
`"ec2:DeleteTags"`
	`"arn:aws:ec2:::network-interface/*", "arn:aws:ec2:::security-group/*"`	Permite criar e excluir tags.
		
		Você precisará dessas permissões se usar um catálogo de AWS Glue dados com uma VPC habilitada. DataBrew passa dados AWS Glue para executar seus trabalhos e projetos. Essas permissões permitem a marcação de recursos do Amazon EC2 criados para endpoints de desenvolvimento.AWS Glue marca interfaces de rede, grupos de segurança e instâncias do Amazon EC2 com. `aws-glue-service-resource`

`"logs:CreateLogGroup"`
 	
`"logs:CreateLogStream"`
 	
`"logs:PutLogEvents"`
	`"arn:aws:logs:::log-group:/aws-glue-databrew/*"`	Permite gravar registros no Amazon CloudWatch Logs
		
		DataBrew grava registros em grupos de registros cujos nomes começam com`aws-glue-databrew` .

Ação	Recurso	Descrição
`"lakeformation:GetDataAccess"`	`"*"`	Permite acesso a AWS Lake Formation, desde `"Glue":"GetTable"` que também seja permitido O uso do Lake Formation requer mais configurações no console do Lake Formation.

Política do IAM para usar objetos do Amazon S3 com DataBrew

A `AwsGlueDataBrewSpecificS3BucketPolicy` política concede as permissões necessárias para acessar o S3 em nome de usuários não administrativos.

Personalize a política da seguinte forma:

1. Substitua os caminhos do Amazon S3 na política para que eles apontem para os caminhos que você deseja usar. No texto de amostra, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* representa um objeto ou arquivo específico. *`BUCKET-NAME-2/`*representa todos os objetos (`*`) cujo nome do caminho começa com`BUCKET-NAME-2/`. Atualize-os para nomear os buckets que você está usando.
2. (Opcional) Use curingas nos caminhos do Amazon S3 para restringir ainda mais as permissões. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

   Prática recomendada de segurança: Para evitar o acesso não autorizado aos buckets do Amazon S3 com nomes semelhantes em AWS outras contas, `aws:ResourceAccount` inclua a chave de condição em sua política. Isso garante que DataBrew você só possa acessar buckets em sua própria AWS conta, mesmo ao usar ARNs de recursos curinga. Adicione a seguinte condição às suas declarações de política:

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

```
}
```

`123456789012`Substitua pelo ID real AWS da sua conta.

Como parte disso, você pode restringir as permissões para as ações `s3:PutObject` `s3:PutBucketCORS` e. Essas ações são necessárias somente para usuários que criam DataBrew projetos, porque esses usuários precisam ser capazes de enviar arquivos de saída para o S3.

Para obter mais informações e ver alguns exemplos do que você pode adicionar a uma política do IAM para o Amazon S3, consulte Exemplos de políticas de bucket no Guia do desenvolvedor do Amazon S3.

A tabela a seguir descreve as permissões concedidas por esta política.

Ação	Recurso	Descrição
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite que você visualize seus arquivos.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite enviar arquivos de saída para o S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite excluir um objeto.

Para definir a política AwsGlueDataBrewSpecificS3BucketPolicy do IAM para DataBrew (console)

1. Baixe o JSON para a política do `AwsGlueDataBrewSpecificS3BucketPolicy`IAM.

2. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, selecione Políticas.
4. Para cada política, escolha Criar política.
5. Na tela Criar política, navegue até a guia JSON.
6. Cole a declaração JSON da política sobre a declaração de amostra no editor.
7. Verifique se a política está personalizada de acordo com sua conta, requisitos de segurança e AWS recursos necessários. Se precisar fazer alterações, você pode fazê-las no editor.
8. Selecione Revisar política.

Para definir a política AwsGlueDataBrewSpecificS3BucketPolicy do IAM para DataBrew (AWS CLI)

1. Baixe o JSON para `AwsGlueDataBrewSpecificS3BucketPolicy`.
2. Personalize a política conforme descrito na primeira etapa do procedimento anterior.
3. Execute o comando a seguir para criar a política.

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

Política do IAM para usar criptografia com DataBrew

A `AwsGlueDataBrewS3EncryptedPolicy` política concede as permissões necessárias para acessar objetos do S3 criptografados com AWS Key Management Service(AWS KMS) em nome de usuários não administrativos.

Personalize a política da seguinte forma:

1. Substitua os caminhos do Amazon S3 na política para que eles apontem para os caminhos que você deseja usar. No texto de amostra, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* representa um objeto ou arquivo específico. *`BUCKET-NAME-2/`*representa todos os objetos (`*`) cujo nome do caminho começa com`BUCKET-NAME-2/`. Atualize-os para nomear os buckets que você está usando.
2. (Opcional) Use curingas nos caminhos do Amazon S3 para restringir ainda mais as permissões. Para obter mais informações, consulte Elementos de política do IAM: variáveis e etiquetas.

Como parte disso, você pode restringir as permissões para as ações `s3:PutObject` `s3:PutBucketCORS` e. Essas ações são necessárias somente para usuários que criam DataBrew projetos, porque esses usuários precisam ser capazes de enviar arquivos de saída para o S3.

Para obter mais informações e ver alguns exemplos do que você pode adicionar a uma política do IAM para o Amazon S3, consulte Exemplos de políticas de bucket.

3. Encontre os seguintes ARNs de recursos no `ToUseKms` arquivo.

```
  "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. Altere a AWS conta de exemplo para o número AWS da sua conta (sem hífens).
5. Altere a lista de amostras para listar as funções do IAM que você deseja usar. Recomendamos definir o escopo de suas políticas do IAM com o menor conjunto de permissões possível. No entanto, você pode permitir que seu usuário acesse todas as funções do IAM, por exemplo, se você estiver usando uma conta de aprendizado pessoal com dados de amostra. Para permitir que a lista acesse todas as funções do IAM, altere a lista de amostra para uma entrada:`"arn:aws:iam::111122223333:role/*"`.

A tabela a seguir descreve as permissões concedidas por esta política.

Ação	Recurso	Descrição
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite que você visualize seus arquivos.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite a listagem de buckets do Amazon S3 a partir de projetos, conjuntos de dados e trabalhos.
`"s3:PutObject"`	`"arn:aws:s3:::bucket_name/*",`	Permite enviar arquivos de saída para o S3.

Ação	Recurso	Descrição
	`"arn:aws:s3:::bucket_name"`	
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite excluir um objeto criado por DataBrew.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Permite a descriptografia de conjuntos de dados criptografados.
`"kms:GenerateDataKey*"`	`"arn:aws:kms:::key/key_ids"`	Permite a criptografia da saída do trabalho.

Para definir a política AwsGlueDataBrewS3EncryptedPolicy do IAM para DataBrew (console)

1. Baixe o JSON para a política do `AwsGlueDataBrewS3EncryptedPolicy`IAM.
2. Faça login no Console de gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, selecione Políticas.
4. Para cada política, escolha Criar política.
5. Na tela Criar política, navegue até a guia JSON.
6. Cole a declaração JSON da política sobre a declaração de amostra no editor.
7. Verifique se a política está personalizada de acordo com sua conta, requisitos de segurança e AWS recursos necessários. Se precisar fazer alterações, você pode fazê-las no editor.
8. Selecione Revisar política.

Para definir a política AwsGlueDataBrewS3EncryptedPolicy do IAM para DataBrew (AWS CLI)

1. Baixe o JSON para `AwsGlueDataBrewS3EncryptedPolicy`.
2. Personalize a política conforme descrito na primeira etapa do procedimento anterior.

3. Execute o comando a seguir para criar a política.

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

Adicionar usuários ou grupos com DataBrew permissões

Você atribui políticas a funções e funções a usuários e grupos para gerenciar permissões. Para obter mais informações, consulte Identidades do IAM (usuários, grupos e funções) no Guia do usuário do IAM.

Antes de começar, você precisa ter pelo menos um usuário ao qual atribuir permissões.

Use o procedimento a seguir para configurar DataBrew permissões para usuários que precisam trabalhar no DataBrew console ou executar DataBrew comandos na CLI.

Para configurar DataBrew permissões

1. Crie uma chave de acesso para que seu usuário use o AWS CLI for DataBrew e outras ferramentas de desenvolvimento.
2. Ative o Console de gerenciamento da AWS acesso para permitir que o usuário use o AWS console.
3. Crie uma função para DataBrew usuários ou grupos.
4. Escolha a política que você está usando. Execute um destes procedimentos:
   - Se você criou`AwsGlueDataBrewCustomUserPolicy`, selecione-o na lista.
   - Para usar a AWS-managed política, `AwsGlueDataBrewFullAccessPolicy` selecione na lista.
5. Atribua essa política à função.
6. Defina as relações de confiança para a função para que um usuário ou grupo possa assumir a função relevante.
   - Se você não estiver usando grupos, confie ao usuário com a função.
   - Se você estiver usando grupos, confie no grupo com a função e adicione o usuário ao grupo.

Adicionar uma função do IAM com permissões de recursos de dados

Você usa funções do IAM para gerenciar políticas que são atribuídas em conjunto. Uma função do IAM pode ser usada por alguém atuando em uma função específica, como um DataBrew usuário ou DataBrew por si mesmo. Para obter mais informações, consulte Perfis do IAM no Guia do usuário do IAM.

Use o procedimento a seguir para criar uma função do IAM necessária para que os DataBrew projetos acessem os dados.

Para anexar a política do IAM necessária a uma nova função do IAM para DataBrew

1. No painel de navegação, selecione Funções e Criar função.
2. Em Selecionar tipo de entidade confiável, escolha o cartão rotulado como AWS serviço.
3. Escolha na DataBrewlista e, em seguida, escolha Avançar: Permissões.
4. Digite **`AwsGlueDataBrewDataResourcePolicy`** na caixa de pesquisa (a política do IAM que você criou em uma etapa anterior). Selecione a política e escolha Avançar: Tags.
5. Escolha Próximo: revisar.
6. Em Nome do perfil, insira **`AwsGlueDataBrewDataAccessRole`** e escolha Criar perfil.

Configurar Centro de Identidade do AWS IAM(Centro de identidade do IAM)

Usando o Centro de Identidade do AWS IAM(IAM Identity Center), seus usuários podem fazer login DataBrew com um URL simples, sem fazer login no Console de gerenciamento da AWS e sem precisar de uma AWS conta.

Para configurar o IAM Identity Center

1. Abra o AWS Organizations console e crie uma organização, caso ainda não tenha uma. Todos os recursos estão habilitados por padrão para essa organização.

   Para obter mais informações, consulte Centro de Identidade do AWS IAM Pré-requisitos e Criação e gerenciamento de uma organização.
2. Abra o console do Centro de Identidade do AWS IAM.
3. Escolha a origem das identidades.

Por padrão, você obtém um armazenamento do Centro de Identidade do IAM para gerenciamento rápido e fácil de usuários. Opcionalmente, você pode conectar um provedor de identidade externo ou conectar um AWS Managed Microsoft AD diretório ao seu Active Directory local. Neste guia, usamos o armazenamento padrão do IAM Identity Center.

Para obter mais informações, consulte Escolha sua fonte de identidade no Guia Centro de Identidade do AWS IAM do usuário.

4. Crie um conjunto de permissões para DataBrew acesso:

   a. No painel de navegação do IAM Identity Center, escolha AWS contas e, em seguida, escolha Conjuntos de permissões.

   b. Na página Criar conjunto de permissões, escolha Criar um conjunto de permissões personalizado.

   c. Em Estado do relé, insira`https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`.

      Inserir isso permite que seus usuários acessem diretamente DataBrew.

   d. Escolha Anexar políticas AWS gerenciadas DataBrew, pesquise e escolha AwsGlueDataBrewFullAccessPolicy. Escolher isso dá aos usuários todas as permissões de que precisam DataBrew. Você pode encontrar mais detalhes emAdicionar uma política do IAM para um usuário do console.

   e. (Opcional) Escolha Criar uma política de permissões personalizada e personalize as permissões para seus usuários.

5. No painel de navegação do Centro de Identidade do IAM, escolha Grupos e escolha Criar grupos. Insira o nome do grupo de escolha Criar.

6. Adicione um usuário à loja do IAM Identity Center:

   a. No painel de navegação do Centro de Identidade do IAM, escolha Usuários.

   b. Na tela Adicionar usuário, insira as informações necessárias e escolha Enviar um e-mail para o usuário com instruções de configuração de senha. O usuário deve receber um e-mail sobre as próximas etapas de configuração.

   c. Selecione Grupos, escolha o grupo ao qual você deseja adicionar o usuário e selecione Adicionar usuários.

Os usuários devem receber um e-mail convidando-os a usar a SSO. Nesse e-mail, eles precisam escolher Aceitar convite e definir a senha. Eles também podem encontrar a URL do portal no e-mail. Eles podem usar esse URL para acessar DataBrew.

7. Atribua cada usuário a uma conta:

   a. Abra o console do IAM Identity Center e, no painel de navegação, escolha AWS contas.

   b. Escolha AWS a organização e escolha uma AWS conta.

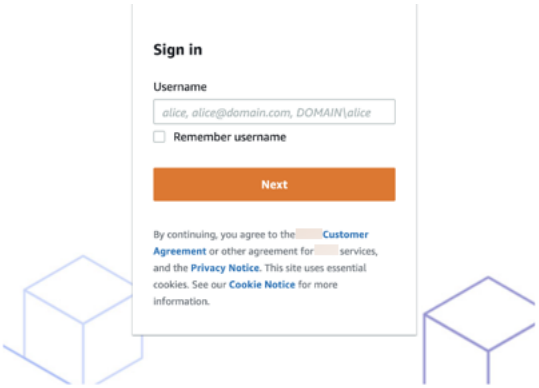
   c. Na tela Atribuir usuários, escolha a guia Grupos e escolha o grupo que você deseja.

   d. Escolha Next: Permission sets (Próximo: conjuntos de permissões).

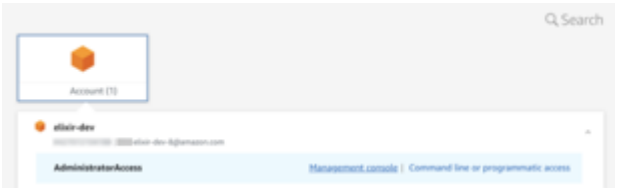
   e. Escolha o conjunto de permissões para DataBrew e escolha Concluir.

Etapas de login para um Center-enabled usuário do IAM Identity

1. Faça login AWS usando uma Center-enabled conta do IAM Identity.



2. Clique em Identidade AWS da conta



3. Clique em Console de gerenciamento para redirecionar com um clique para o DataBrew console.

Usando DataBrew como uma extensão em JupyterLab

> **Warning**
>
> AWS Glue DataBrew JupyterLab o suporte de extensão terminará em 31 de dezembro de 2024, pois JupyterLab 3 chegarão ao fim do suporte. Para obter mais informações, consulte JupyterLab 3. Fim da manutenção.

Se você preferir preparar dados em um ambiente Jupyter Notebook, você pode usar todos os recursos do AWS Glue DataBrew in. JupyterLab

JupyterLab é um ambiente de desenvolvimento interativo baseado na web para o Jupyter Notebook. Na JupyterLab página da Web local, você pode adicionar seções para um terminal, uma sessão SQL, Python e muito mais. Depois de instalar a AWS Glue DataBrew extensão, você pode adicionar uma seção para o DataBrew console. Ele funciona com qualquer notebook existente ou outras extensões que você já tenha, diretamente do JupyterLab ambiente.

Tópicos

Pré-requisitos

Antes de começar, configure os seguintes itens:

- Uma AWS conta — Se você ainda não tem uma, comece comConfigurando um novo AWS account.
- Um usuário AWS Identity and Access Management(IAM) com acesso às permissões necessárias para DataBrew — Para obter mais informações, consulteAdicionar usuários ou grupos com DataBrew permissões.
- Uma função do IAM para usar em DataBrew operações — você pode usar o padrão, se `AwsGlueDataBrewDataAccessRole` estiver configurado. Para configurar funções adicionais do IAM, consulteAdicionar uma função do IAM com permissões de recursos de dados.

- Uma JupyterLab instalação (versão 2.2.6 ou superior) — Para obter mais informações, consulte os seguintes tópicos na JupyterLabdocumentação:
  - JupyterLab Pré-requisitos
  - JupyterLab instalação — Recomendamos o usopip install jupyterlab.
- Uma Node.js instalação (versão 12.0 ou superior).
- Uma instalação AWS Command Line Interface(AWS CLI) — Para obter mais informações, consulteConfigurando o AWS CLI.
- Uma instalação de proxy AWS Jupyter (pip install aws-jupyter-proxy) — Essa extensão é usada com um endpoint de AWS serviço para passar suas credenciais com segurança.AWS Para obter mais informações, consulte aws-jupyter-proxy em. GitHub

Para verificar se você tem os pré-requisitos instalados, você pode executar um teste semelhante ao seguinte na linha de comando, conforme mostrado no exemplo a seguir.

```
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
```

A saída deve parecer com algo semelhante ao seguinte: Os diretórios variam de acordo com o sistema operacional e a configuração.

```
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
```

Configurando JupyterLab para usar a extensão

Depois de instalar JupyterLab, você precisa configurá-lo para proteger o acesso aos dados e habilitar extensões de servidor.

Para configurar uma senha e criptografia

1. Defina uma senha para proteger os dados que você planeja adicionar à extensão. O Jupyter fornece um utilitário de senha. Execute o comando a seguir e insira sua senha preferencial no prompt.

   ```
   jupyter notebook password
   ```

   A saída é semelhante ao apontado abaixo.

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

2. Ative a criptografia no servidor Jupyter. Se você instalar o Jupyter em sua máquina local e ninguém puder acessá-lo pela rede, você pode pular esta etapa.

   Para configurar a criptografia com o Transport Layer Security (TLS), crie um certificado personalizado para seu ambiente. Para obter mais informações, Como usar o Let's Encrypt na proteção de um servidor na documentação do Jupyter.

3. Para começar JupyterLab, execute o comando a seguir no prompt de comando.

```
jupyter lab
```

Para obter mais informações, consulte Iniciando JupyterLab na JupyterLab documentação.

4. Enquanto JupyterLab estiver executando, você pode acessá-lo em um URL semelhante ao seguinte: http://localhost:8888/lab. Se você configurar a criptografia, use `https` em vez de`http`. Se você personalizou a porta, substitua o número da porta em vez de`8888`.

Use o procedimento a seguir para ativar as extensões de terceiros.

Para habilitar extensões de terceiros no JupyterLab

1. Na JupyterLab página da web, escolha o ícone do Extension Manager no menu à esquerda.
2. Leia o aviso sobre os riscos de executar extensões de terceiros. Instale somente extensões de desenvolvedores em quem você confia.
3. Para ativar extensões de terceiros JupyterLab, escolha Ativar.
4. Siga as instruções para reconstruir e recarregar. JupyterLab

Ativando a DataBrew extensão para JupyterLab

Depois de ter uma instalação segura JupyterLab com as extensões ativadas, instale a DataBrew extensão para que você possa executá-la DataBrew em seu notebook.

Para instalar as extensões para DataBrew (console)

1. Para começar JupyterLab, execute o comando a seguir no prompt de comando.

```
jupyter lab
```

2. Na JupyterLab página da web, escolha o ícone do Extension Manager no menu à esquerda.
3. Pesquise a DataBrew extensão digitando "**`brew`**" em Pesquisar no canto superior esquerdo.
4. Localize aws_glue_databrew_jupyter na lista, mas não clique nela. Se você clicar no nome destacado da extensão, uma nova janela do navegador será aberta com a página aws_glue_databrew_jupyter ativada. GitHub
5. Para instalar a DataBrew extensão, escolha uma das seguintes opções:

- Na linha de comando, execute`jupyter labextension install aws_glue_databrew_jupyter`.
- Escolha Instalar na parte inferior da placa de extensão, abaixo de "aws_glue_databrew_jupyter" em letras cinza.

DataBrew a extensão é compatível com as JupyterLab versões 1.2 e 2.x.

6. Para verificar se ele está instalado, execute`jupyter labextension list`. A saída deve parecer com algo semelhante ao seguinte:

```
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. Reconstrua JupyterLab usando uma das seguintes opções:
   - No prompt de comando, execute`jupyter lab build`.
   - Na página da web, escolha Reconstruir no canto superior esquerdo.
8. Quando a compilação estiver concluída, faça o seguinte:
   - No prompt de comando, execute`jupyter lab`.
   - Na página da web, escolha Recarregar na mensagem Build Complete.
9. Na JupyterLab página da web, feche o Extension Manager escolhendo seu ícone no menu à esquerda.

   Para abrir a extensão, escolha Iniciar AWS Glue DataBrew na seção Outros na guia Inicializador. A extensão usa sua AWS CLI configuração atual para chaves de acesso e configurações de AWS região.

Depois de concluir a configuração, você pode usar a AWS Glue DataBrewguia para interagir DataBrew internamente JupyterLab.

Conceitos básicos de AWS Glue DataBrew

Você pode usar o tutorial a seguir para orientá-lo na criação do seu primeiro DataBrew projeto. Você carrega um conjunto de dados de amostra, executa transformações nesse conjunto de dados, cria uma receita para capturar essas transformações e executa um trabalho para gravar os dados transformados no Amazon S3.

Tópicos

Pré-requisitos

Antes de continuar, siga as instruções aplicáveis emConfigurar AWS Glue DataBrew. Em seguida, continueEtapa 1: criar um projeto.

Etapa 1: criar um projeto

Nesta etapa, você usa o DataBrew console para começar rapidamente com um projeto de amostra.

Para criar um projeto

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console em https://console.aws.amazon.com/databrew/.
2. Certifique-se de que sua AWS região esteja selecionada no canto superior direito do DataBrew console. Para obter uma lista das AWS regiões suportadas pelo DataBrew, consulte DataBrew endpoints e cotas no.Referência geral da AWS
3. No painel de navegação, escolha Projetos e, em seguida, escolha Criar projeto.

4. No painel Detalhes do projeto, faça o seguinte:

   - Em Nome do projeto, insira`chess-project`.
   - Para Receita anexada, crie uma nova receita. Um nome sugerido para a receita é fornecido (`chess-project-recipe`).

5. No painel Selecionar um conjunto de dados, escolha Arquivos de amostra.
6. No painel Arquivos de amostra, escolha Movimentos famosos de jogos de xadrez. Esse conjunto de dados contém informações detalhadas sobre mais de 20.000 jogos de xadrez.

   Para o nome do conjunto de dados, um nome sugerido para o conjunto de dados é fornecido ()`chess-games`.
7. No painel Permissões de acesso, escolha`AwsGlueDataBrewDataAccessRole`. Essa é uma função vinculada ao serviço que permite DataBrew acessar seus buckets do Amazon S3 em seu nome.
8. Escolha Criar projeto e espere até DataBrew terminar de preparar o projeto. A janela é semelhante à seguinte.

   Os dados que você vê representam uma amostra do `chess-games` conjunto de dados. Por padrão, a amostra consiste nas primeiras 500 linhas do conjunto de dados. Você pode alterar essa configuração do projeto posteriormente.

   A barra de ferramentas fornece acesso a centenas de transformações de dados que você pode aplicar aos dados.

   O painel de receitas à direita no DataBrew console rastreia as transformações que você aplicou até o momento.

Etapa 2: resumir os dados

Nesta etapa, você cria uma DataBrew receita — um conjunto de transformações que podem ser aplicadas a esse conjunto de dados e a outros semelhantes. Quando a receita estiver completa, você a publica para que fique disponível para uso.

No jogo de xadrez, os jogadores podem ser avaliados com base em seu desempenho contra outros jogadores. (Para obter mais informações, consulte https://en.wikipedia.org/wiki/Chess_rating_system). Neste tutorial, você se concentra apenas nos jogos em que os dois jogadores eram da Classe A, o que significa que suas classificações foram de 1800 ou mais.

Para resumir os dados

1. Na barra de ferramentas de transformação, escolha Filtrar, Por condição, Maior ou igual a.
2. Defina essas opções da seguinte forma:

   - Coluna de origem - `white_rating`
   - Condição do filtro — Maior ou igual a 1800

   Para ver como a transformação funciona, escolha Visualizar alterações. Em seguida, escolha Aplicar.
3. Repita a etapa anterior, mas desta vez defina a coluna Fonte como`black_rating`. Depois de aplicar suas alterações, os dados de amostra contêm somente os jogos em que os jogadores de cada lado (preto e branco) eram da Classe A ou superior.
4. Resuma os dados para determinar quantos jogos foram vencidos por cada lado. Para fazer isso, na barra de ferramentas de transformação, escolha Grupo.
5. Para as propriedades do Grupo, faça o seguinte:

   a. Na primeira linha, escolha o `winner` nome da coluna. Deixe Agregado definido como Agrupar por.
   b. Na segunda linha, escolha `victory_status` o nome da coluna. Deixe Agregado definido como Agrupar por.
   c. Escolha Adicionar outra coluna.
   d. Na terceira linha, escolha o `winner` nome da coluna. Defina Agregar como Contagem.
   e. Em Tipo de grupo, escolha Grupo como nova tabela. O painel de visualização mostra a aparência do resultado.
   f. Escolha Terminar.
6. Escolha Publicar para salvar seu trabalho, à direita no painel de receitas.
7. Em Descrição da versão, insira Primeira versão da minha receita. Em seguida, escolha Publicar.

Etapa 3: adicionar mais transformações

Nesta etapa, você adiciona mais transformações à sua receita e publica outra versão dela. Para refinar nosso exemplo, usamos a informação de que nem todos os jogos de xadrez resultam em um vencedor claro; alguns jogos são disputados até o empate.

Para adicionar mais transformações de receita e republicar

1. Na barra de ferramentas de transformação, escolha Filtrar, Por condição, É para não remover os jogos que foram jogados até o empate.
2. Defina essas opções da seguinte forma:

   - Coluna de origem - `victory_status`
   - Condição do filtro — Não é `draw`

   Para adicionar essa transformação à sua receita, escolha Aplicar.
3. Altere os dados `victory_status` para que sejam mais significativos. Para fazer isso, na barra de ferramentas de transformação, escolha Limpar, Substituir, Substituir valor ou padrão.
4. Defina essas opções da seguinte forma:

   - Coluna de origem - `victory_status`
   - Especificar valores a serem substituídos — Valor ou padrão
   - Valor a ser substituído - `mate`
   - Substituir por valor - `checkmate`

   Para adicionar essa transformação à sua receita, escolha Aplicar.
5. Repita a etapa anterior, mas `resign` mude para`other player resigned`.
6. Repita a etapa anterior, mas `outoftime` mude para`time ran out`.
7. Escolha Publicar para salvar seu trabalho, à direita no painel de receitas.

Etapa 4: revise seus DataBrew recursos

Agora que você trabalhou com um projeto de amostra, revise os DataBrew recursos que você criou até agora.

Para revisar seus DataBrew recursos

1. No painel de navegação, escolha Conjuntos de dados.

   Quando você criou o projeto de amostra, DataBrew criou um conjunto de dados para você (`chess-games`). O arquivo de dados de origem é armazenado no Amazon S3 e está no formato

Microsoft Excel ()`chess-games.xlsx`. O arquivo contém metadados de mais de 20.000 jogos de xadrez. O `chess-games` conjunto de dados fornece as informações DataBrew necessárias para ler os dados nesse arquivo.

2. No painel de navegação, escolha Projetos.

   Você deve ver o projeto com o qual trabalhou nas etapas anteriores (`chess-project`). Todo projeto requer um conjunto de dados, nesse caso`chess-games`. Cada projeto também exige uma receita, para que você possa adicionar etapas de transformação de dados à medida que avança. Ao criar esse projeto de amostra, DataBrew criou uma nova receita (vazia) para você e a anexou ao projeto.

3. No painel de navegação, escolha Receitas e, na coluna Nome da receita, escolha chess-project-recipe. Isso mostra a receita DataBrew criada para seu projeto e que você refinou adicionando etapas de transformação a ela.
4. À esquerda, veja as versões da receita que foram publicadas. Escolha uma delas para ver a guia Etapas da receita, que mostra os detalhes da receita e as etapas dessa versão.
5. Veja a guia Linhagem de dados, que mostra de onde os dados vieram e como estão sendo usados. Para obter mais detalhes, escolha qualquer um dos ícones no diagrama.

Etapa 5: criar um perfil de dados

Quando você trabalha em um projeto, DataBrew exibe estatísticas como o número de linhas na amostra e a distribuição de valores exclusivos em cada coluna. Essas estatísticas, e muitas outras, representam um perfil da amostra.

Para solicitar um perfil de dados, crie e execute um trabalho de perfil.

Para criar o perfil de um conjunto de dados

1. No painel de navegação, escolha Trabalhos.
2. Na guia Perfil de trabalhos, escolha Criar trabalho.
3. Em Nome do trabalho, insira`chess-data-profile`.
4. Em Tipo de trabalho, escolha Criar um trabalho de perfil.
5. No painel Job input, faça o seguinte:

   - Em Executar em, escolha Conjunto de dados.

- Escolha Selecionar um conjunto de dados para ver uma lista dos conjuntos de dados disponíveis e escolha. `chess-games`

6. No painel Configurações de saída do Job, faça o seguinte:

   - Em Tipo de arquivo, escolha JSON (notação de JavaScript objeto).
   - Escolha a localização do S3 para ver uma lista de buckets Amazon S3 disponíveis e escolha o bucket a ser usado. Em seguida, selecione Procurar. Na lista de pastas`databrew-output`, escolha e escolha Selecionar.

7. No painel Permissões de acesso, escolha`AwsGlueDataBrewDataAccessRole`. Essa é uma função vinculada ao serviço que permite DataBrew acessar seus buckets do Amazon S3 em seu nome.
8. Escolha Criar e executar tarefa. DataBrew cria um trabalho com suas configurações e o executa.
9. No painel Histórico de execução de trabalhos, aguarde até que o status do trabalho mude de `Running` para`Succeeded`.
10. Para ver o perfil, escolha VER PERFIL:

    A janela DATASETS é exibida. Reserve um tempo para explorar as seguintes guias:

    - Pré-visualização do conjunto de dados
    - Visão geral do perfil
    - Estatísticas de colunas
    - Estatísticas de linhagem de dados

Etapa 6: transformar o conjunto de dados

Até agora, você testou sua receita somente em uma amostra do conjunto de dados. Agora é hora de transformar todo o conjunto de dados criando um trabalho de DataBrew receita.

Quando o trabalho é executado, DataBrew aplica sua receita a todos os dados no conjunto de dados e grava os dados transformados em um bucket do Amazon S3. Os dados transformados são separados do conjunto de dados original. DataBrew não altera os dados de origem.

Antes de continuar, certifique-se de ter um bucket do Amazon S3 em sua conta no qual você possa gravar. Nesse bucket, crie uma pasta para capturar a saída do trabalho DataBrew. Para executar essas etapas, use o procedimento a seguir.

Para criar um bucket e uma pasta do S3 para capturar a saída do trabalho

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon S3 em. https://console.aws.amazon.com/databrew/

   Se você já tem um bucket do Amazon S3 disponível e tem permissões de gravação para ele, pule a próxima etapa.
2. Se você não tiver um bucket do Amazon S3, escolha Create bucket. Em Nome do bucket, insira um nome exclusivo para seu novo bucket. Selecione Criar bucket.
3. Na lista de compartimentos, escolha aquele que você deseja usar.
4. Selecione Criar pasta.
5. Em Nome da pasta`databrew-output`, insira e escolha Criar pasta.

Depois de criar um bucket e uma pasta do Amazon S3 para conter o trabalho, execute seu trabalho usando o procedimento a seguir.

Para criar e executar um trabalho de receita

1. No painel de navegação, escolha Trabalhos.
2. Na guia Tarefas de receita, escolha Criar tarefa.
3. Em Nome do trabalho, insira`chess-winner-summary`.
4. Em Tipo de trabalho, escolha Criar um trabalho de receita.
5. No painel Job input, faça o seguinte:

   - Em Executar em, escolha Conjunto de dados.
   - Escolha Selecionar um conjunto de dados para ver uma lista dos conjuntos de dados disponíveis e escolha. `chess-games`
   - Escolha Selecionar uma receita para ver uma lista de receitas disponíveis e escolha`chess-project-recipe`.
6. No painel Configurações de saída do Job, faça o seguinte:

   - Tipo de arquivo — escolha CSV (valores separados por vírgula).

- Localização do S3 - escolha esse campo para ver uma lista de buckets Amazon S3 disponíveis e escolha o bucket a ser usado. Em seguida, selecione Procurar. Na lista de pastas`databrew-output`, escolha e escolha Selecionar.

7. No painel Permissões de acesso, escolha`AwsGlueDataBrewDataAccessRole`. Essa função vinculada ao serviço permite DataBrew acessar seus buckets do Amazon S3 em seu nome.
8. Escolha Criar e executar tarefa. DataBrew cria um trabalho com suas configurações e o executa.
9. No painel Histórico de execução de trabalhos, aguarde até que o status do trabalho mude de `Running` para`Succeeded`.
10. Escolha Saída para acessar o console do Amazon S3. Escolha seu bucket do S3 e, em seguida, escolha a `databrew-output` pasta para acessar a saída do trabalho.
11. (Opcional) Escolha Baixar para baixar o arquivo e visualizar seu conteúdo.

Etapa 7: (opcional) limpeza

O passo a passo está completo. Você pode continuar usando DataBrew os recursos do Amazon S3 que você criou ou excluí-los.

Para limpar os recursos

1. Abra o DataBrew console em e https://console.aws.amazon.com/databrew/, no painel de navegação, escolha Projetos.
2. Escolha seu projeto (Projeto de amostra). Em Ações, escolha Excluir.
3. No painel Excluir amostra do projeto, escolha Excluir receita anexada. Escolha Excluir. Seu projeto, junto com sua receita e trabalhos, serão excluídos.
4. No painel de navegação, escolha Conjuntos de dados.
5. Escolha seu conjunto de dados (`chess-games`) e, em Ações, escolha Excluir.
6. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/. Exclua a `databrew-output` pasta e seu conteúdo.

   (Opcional) Se tiver certeza de que não precisa mais do seu bucket do Amazon S3, você pode excluí-lo.

Conectando-se aos dados com AWS Glue DataBrew

Em AWS Glue DataBrew, um conjunto de dados representa dados que são carregados de um arquivo ou armazenados em outro lugar. Por exemplo, os dados podem ser armazenados no Amazon S3, em uma fonte de dados JDBC compatível ou em um catálogo de dados.AWS Glue Se você não estiver carregando um arquivo diretamente para DataBrew, o conjunto de dados também contém detalhes sobre como se DataBrew conectar aos dados.

Ao criar seu conjunto de dados (por exemplo,`inventory-dataset`), você insere os detalhes da conexão somente uma vez. A partir desse ponto, DataBrew pode acessar os dados subjacentes para você. Com essa abordagem, você pode criar projetos e desenvolver transformações para seus dados, sem precisar se preocupar com detalhes de conexão ou formatos de arquivo.

Tópicos

Tipos de arquivo compatíveis com fontes de dados

Os requisitos de arquivo a seguir se aplicam aos arquivos armazenados no Amazon S3 e aos arquivos que você carrega de uma unidade local. DataBrew suporta os seguintes formatos de arquivo: valor separado por vírgula (CSV), Microsoft Excel, JSON, ORC e Parquet. Você pode usar arquivos com uma extensão não padrão ou sem extensão se o arquivo for de um dos tipos suportados.

Se não DataBrew for possível inferir o tipo de arquivo, certifique-se de selecionar você mesmo o tipo de arquivo correto (CSV, Excel, JSON, ORC ou Parquet). Há suporte para arquivos CSV, JSON, ORC e Parquet compactados, mas os arquivos CSV e JSON devem incluir o codec de compactação

como extensão do arquivo. Se você estiver importando uma pasta, todos os arquivos na pasta deverão ser do mesmo tipo de arquivo.

Os formatos de arquivo e os algoritmos de compactação compatíveis são mostrados na tabela a seguir.

> **Note**
>
> Arquivos CSV, Excel e JSON devem ser codificados com Unicode (). UTF-8

Formato	Extensão do arquivo (opcional)	Extensões para arquivos compactados (obrigatório)

Comma-separated valores	`.csv`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Pasta de trabalho do Microsoft Excel	`.xlsx`	Sem suporte para compressão

JSON (documento JSON e linhas JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

Formato	Extensão do arquivo (opcional)	Extensões para arquivos compactados (obrigatório)

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

Conexões suportadas para fontes e saídas de dados

Você pode se conectar às seguintes fontes de dados para tarefas de DataBrew receitas. Isso inclui qualquer fonte de dados que não seja um arquivo para DataBrew o qual você está enviando diretamente. A fonte de dados que você está usando pode ser chamada de banco de dados, data warehouse ou outra coisa. Nós nos referimos a todos os provedores de dados como fontes de dados ou conexões.

Você pode criar um conjunto de dados usando qualquer um dos itens a seguir como fontes de dados.

Você também pode usar bancos de dados Amazon S3 ou JDBC suportados pelo Amazon RDS para a saída de trabalhos de receitas.AWS Glue Data Catalog DataBrew Amazon AppFlow e AWS Data Exchange não são suportados armazenamentos de dados para a produção de trabalhos de DataBrew receitas.

- Amazon S3

  Você pode usar o S3 para armazenar e proteger qualquer quantidade de dados. Para criar um conjunto de dados, você especifica uma URL do S3 onde DataBrew pode acessar um arquivo de dados, por exemplo: `s3://your-bucket-name/inventory-data.csv`

  DataBrew também pode ler todos os arquivos em uma pasta do S3, o que significa que você pode criar um conjunto de dados que abranja vários arquivos. Para fazer isso, especifique um URL do S3 neste formato:`s3://your-bucket-name/your-folder-name/`.

  DataBrew suporta somente as seguintes classes de armazenamento do Amazon S3: Standard, Reduced Redundancy e S3 One Standard-IA. Zone-IA DataBrew ignora arquivos com outras classes de armazenamento. DataBrew também ignora arquivos vazios (arquivos contendo 0 bytes). Para obter mais informações sobre as classes de armazenamento do Amazon S3, consulte

Usando as classes de armazenamento do Amazon S3 no Guia do usuário do console do Amazon S3.

- AWS Glue Data Catalog

  Você pode usar o Catálogo de Dados para definir referências aos dados armazenados na AWS nuvem. Com o Catálogo de Dados, você pode criar conexões com tabelas individuais nos seguintes serviços:

  - Catálogo de dados Amazon S3
  - Catálogo de dados Amazon Redshift
  - Catálogo de dados Amazon RDS
  - AWS Glue

  DataBrew também pode ler todos os arquivos em uma pasta do Amazon S3, o que significa que você pode criar um conjunto de dados que abrange vários arquivos. Para fazer isso, especifique uma URL do Amazon S3 neste formato: `s3://your-bucket-name/your-folder-name/`

  Para serem usadas com DataBrew, as tabelas do Amazon S3 definidas no AWS Glue Data Catalog, devem ter uma propriedade de tabela adicionada a elas chamada `classification`, que identifica o formato dos dados como`csv`, `jsonparquet`, ou e como. `typeOfData file` Se a propriedade da tabela não tiver sido adicionada quando a tabela foi criada, você poderá adicioná-la usando o AWS Glue console.

  DataBrew suporta somente as classes de armazenamento Standard, Reduced Redundancy e S3 One do Amazon S3. Standard-IA Zone-IA DataBrew ignora arquivos com outras classes de armazenamento. DataBrew também ignora arquivos vazios (arquivos contendo 0 bytes). Para obter mais informações sobre as classes de armazenamento do Amazon S3, consulte Usando as classes de armazenamento do Amazon S3 no Guia do usuário do console do Amazon S3.

  DataBrew também pode acessar tabelas do AWS Glue Data Catalog S3 de outras contas se uma política de recursos apropriada for criada. Você pode criar uma política no AWS Glue console na guia Configurações em Catálogo de dados. A seguir está um exemplo de política específica para um único Região da AWS.

  > **Warning**
  >
  > Esta é uma política de recursos altamente permissiva que concede acesso * `$ACCOUNT_TO*` irrestrito ao Catálogo de Dados de. *`$ACCOUNT_FROM*` Na maioria dos

casos, recomendamos que você restrinja sua política de recursos a catálogos ou tabelas específicos. Para obter mais informações, consulte as políticas de AWS Glue recursos para controle de acesso no Guia do AWS Glue desenvolvedor.

Em alguns casos, talvez você queira criar um projeto ou executar um trabalho *$ACCOUNT_TO* com uma tabela do Catálogo de AWS Glue Dados do S3 *$ACCOUNT_FROM* que aponta para um local do S3 que também está em.AWS Glue DataBrew*$ACCOUNT_FROM* Nesses casos, a função do IAM usada ao criar o projeto e o trabalho em *$ACCOUNT_TO* deve ter permissão para listar e obter objetos desse local do *$ACCOUNT_FROM* S3. Para obter mais informações, consulte Conceder acesso entre contas no Guia do AWS Glue desenvolvedor.

- Dados conectados usando drivers JDBC

  Você pode criar um conjunto de dados conectando-se aos dados com um driver JDBC compatível. Para obter mais informações, consulte Usando drivers com AWS Glue DataBrew.

  DataBrew suporta oficialmente as seguintes fontes de dados usando Java Database Connectivity (JDBC):

  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - banco de dados de origem
  - Conector Snowflake para Spark

  As fontes de dados podem estar localizadas em qualquer lugar de onde você possa se conectar a elas DataBrew. Essa lista inclui somente conexões JDBC que testamos e, portanto, podemos oferecer suporte.

  As fontes de dados do Amazon Redshift e do Snowflake Connector para Spark podem ser conectadas de uma das seguintes formas:

  - Com um nome de tabela.
  - Com uma consulta SQL que abrange várias tabelas e operações.

  As consultas SQL são executadas quando você inicia um projeto ou a execução de um trabalho.

Para se conectar a dados que exijam um driver JDBC não listado, verifique se o driver é compatível com o JDK 8. Para usar o driver, armazene-o no S3 em um bucket onde você possa acessá-lo com sua função do IAM. DataBrew Em seguida, aponte seu conjunto de dados para o arquivo do driver. Para obter mais informações, consulte Usando drivers com AWS Glue DataBrew.

Exemplo de consulta para um SQL-based conjunto de dados:

```
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
```

Limitações do SQL personalizado

Se você usa uma conexão JDBC para acessar dados de um DataBrew conjunto de dados, lembre-se do seguinte:

- AWS Glue DataBrew não valida o SQL personalizado que você fornece como parte da criação do conjunto de dados. A consulta SQL será executada quando você iniciar um projeto ou execução de um trabalho. DataBrew pega a consulta fornecida e a passa para o mecanismo de banco de dados usando os drivers JDBC padrão ou fornecidos.
- Um conjunto de dados criado com uma consulta inválida falhará quando for usado em um projeto ou trabalho. Valide sua consulta antes de criar o conjunto de dados.
- O recurso Validate SQL está disponível somente para fontes de Redshift-based dados da Amazon.
- Se você quiser usar um conjunto de dados em um projeto, limite o tempo de execução da consulta SQL a menos de três minutos para evitar um tempo limite durante o carregamento do projeto. Verifique o tempo de execução da consulta antes de criar um projeto.

- Amazon AppFlow

Usando a Amazon AppFlow, você pode transferir dados para o Amazon S3 a partir de aplicativos de terceiros (Software-as-a-Service SaaS), como Salesforce, Zendesk, Slack e. ServiceNow Em seguida, você pode usar os dados para criar um DataBrew conjunto de dados.

Na Amazon AppFlow, você cria uma conexão e um fluxo para transferir dados entre seu aplicativo de terceiros e um aplicativo de destino. Ao usar a Amazon AppFlow com DataBrew, certifique-se de que o aplicativo de AppFlow destino da Amazon seja o Amazon S3. Aplicativos de AppFlow destino da Amazon que não sejam o Amazon S3 não aparecem no DataBrew console. Para obter mais informações sobre a transferência de dados de seu aplicativo de terceiros e a criação de AppFlow conexões e fluxos da Amazon, consulte a AppFlow documentação da Amazon.

Quando você escolhe Connect new dataset na guia Datasets DataBrew e clica em Amazon AppFlow, você vê todos os fluxos na Amazon AppFlow que estão configurados com o Amazon S3 como aplicativo de destino. Para usar os dados de um fluxo para seu conjunto de dados, escolha esse fluxo.

Escolher Criar fluxo, Gerenciar fluxos e Visualizar detalhes da Amazon AppFlow no DataBrew console abre o AppFlow console da Amazon para que você possa realizar essas tarefas.

Depois de criar um conjunto de dados da Amazon AppFlow, você pode executar o fluxo e visualizar os detalhes mais recentes da execução do fluxo ao visualizar os detalhes do conjunto de dados ou do trabalho. Quando você executa o fluxo DataBrew, o conjunto de dados é atualizado no S3 e está pronto para ser usado. DataBrew

As seguintes situações podem surgir quando você seleciona um AppFlow fluxo da Amazon no DataBrew console para criar um conjunto de dados:

- Os dados não foram agregados — se o gatilho do fluxo for Executar sob demanda ou for Executado dentro do cronograma com transferência total de dados, certifique-se de agregar os dados do fluxo antes de usá-los para criar um DataBrew conjunto de dados. A agregação do fluxo combina todos os registros no fluxo em um único arquivo. Fluxos com o tipo de acionador Executar conforme agendamento com transferência incremental de dados ou Executar em evento não exigem agregação. Para agregar dados na Amazon AppFlow, escolha Editar configuração de fluxo > Detalhes do destino > Configurações adicionais > Preferência de transferência de dados.
- O fluxo não foi executado - se o status de execução de um fluxo estiver vazio, isso significa um dos seguintes:
  - Se o gatilho para executar o fluxo for Executar sob demanda, o fluxo ainda não foi executado.

- Se o gatilho para executar o fluxo for Executar no evento, o evento acionador ainda não ocorreu.
- Se o gatilho para executar o fluxo for Executar dentro do cronograma, uma execução programada ainda não ocorreu.

Antes de criar um conjunto de dados com um fluxo, escolha Executar fluxo para esse fluxo.

Para obter mais informações, consulte AppFlow os fluxos da Amazon no Guia AppFlow do usuário da Amazon.

- AWS Data Exchange

Você pode escolher entre centenas de fontes de dados de terceiros que estão disponíveis em AWS Data Exchange. Ao assinar essas fontes de dados, você obtém a versão mais atualizada dos dados.

Para criar um conjunto de dados, você especifica o nome de um produto de AWS Data Exchange dados que você está inscrito e tem o direito de usar.

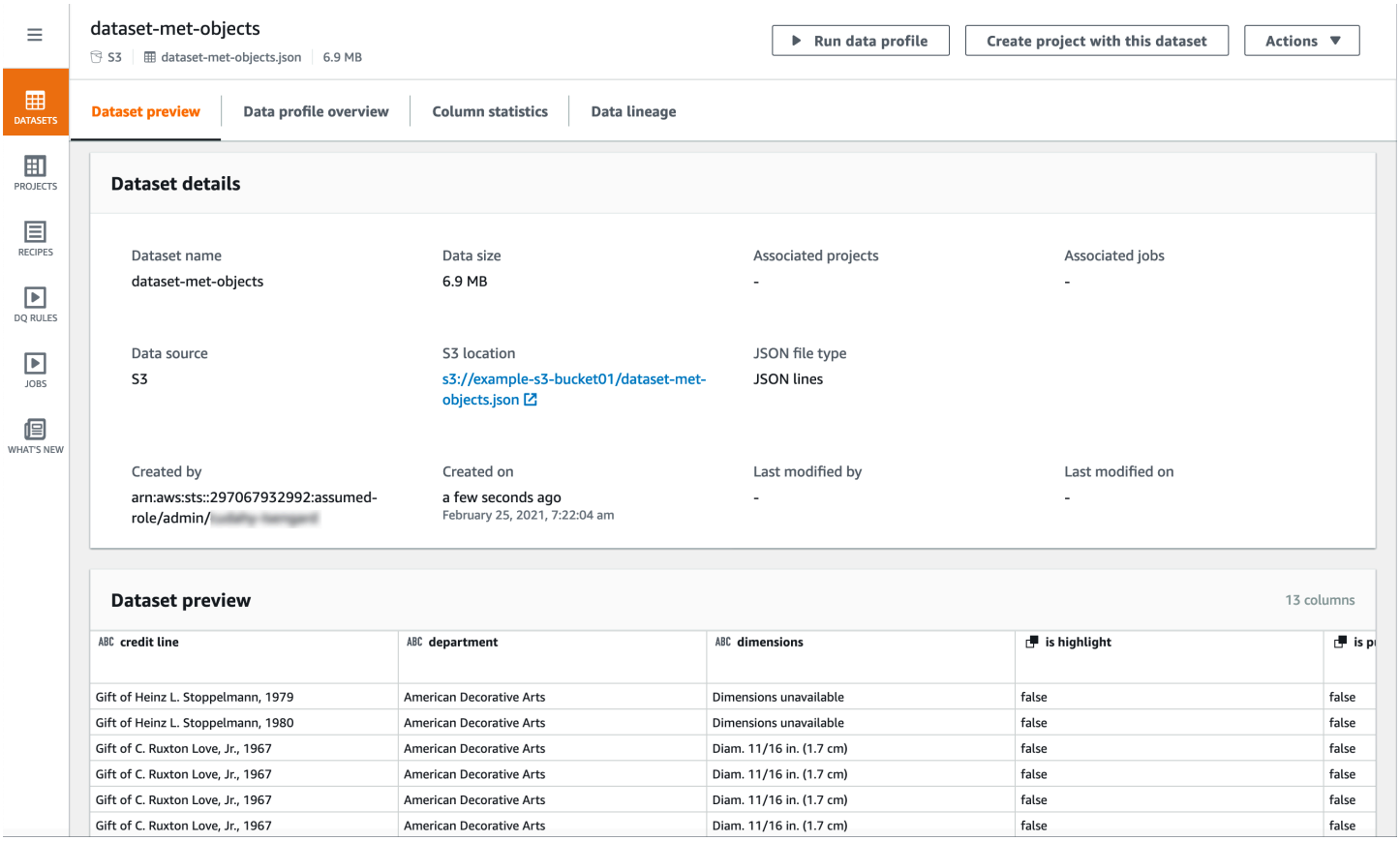
Usando conjuntos de dados em AWS Glue DataBrew

Para ver uma lista dos seus conjuntos de dados no DataBrew console, escolha DATASET à esquerda. Na página de conjuntos de dados, você pode visualizar informações detalhadas de cada conjunto de dados clicando em seu nome ou escolhendo Ações, Editar no menu de contexto.

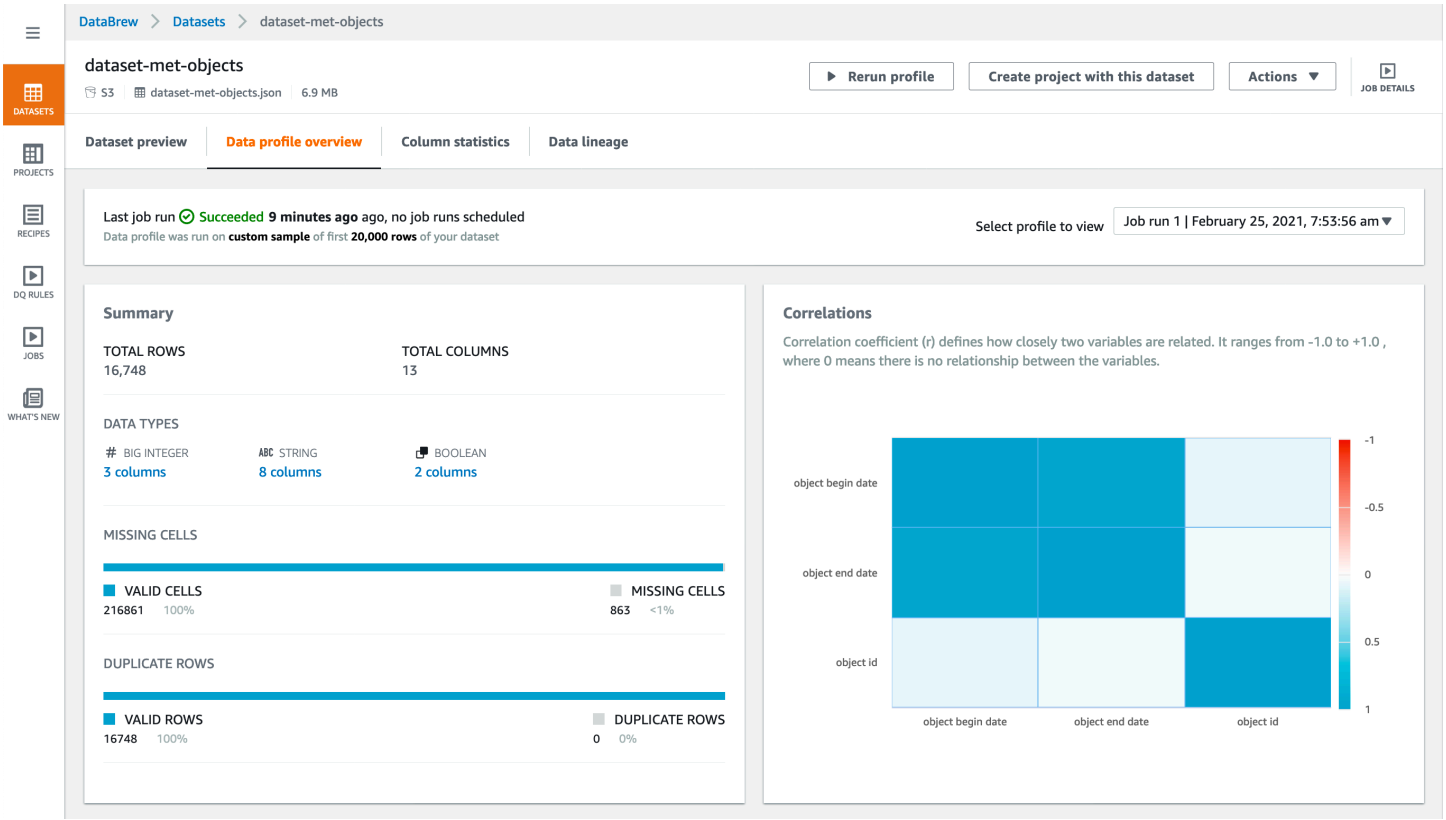
Para criar um novo conjunto de dados, você escolhe DATASET, Connect new dataset. Fontes de dados diferentes têm parâmetros de conexão diferentes e você os insere para que DataBrew possa se conectar. Quando você salva sua conexão e escolhe Criar conjunto de dados, DataBrew se conecta aos seus dados e começa a carregar os dados. Para obter mais informações, consulte Conectando-se aos seus dados.

A página do conjunto de dados tem os seguintes elementos para ajudá-lo a explorar seus dados.

Visualização do conjunto de dados — Nessa guia, você pode encontrar informações de conexão para o conjunto de dados e uma visão geral da estrutura geral do conjunto de dados, conforme mostrado a seguir.

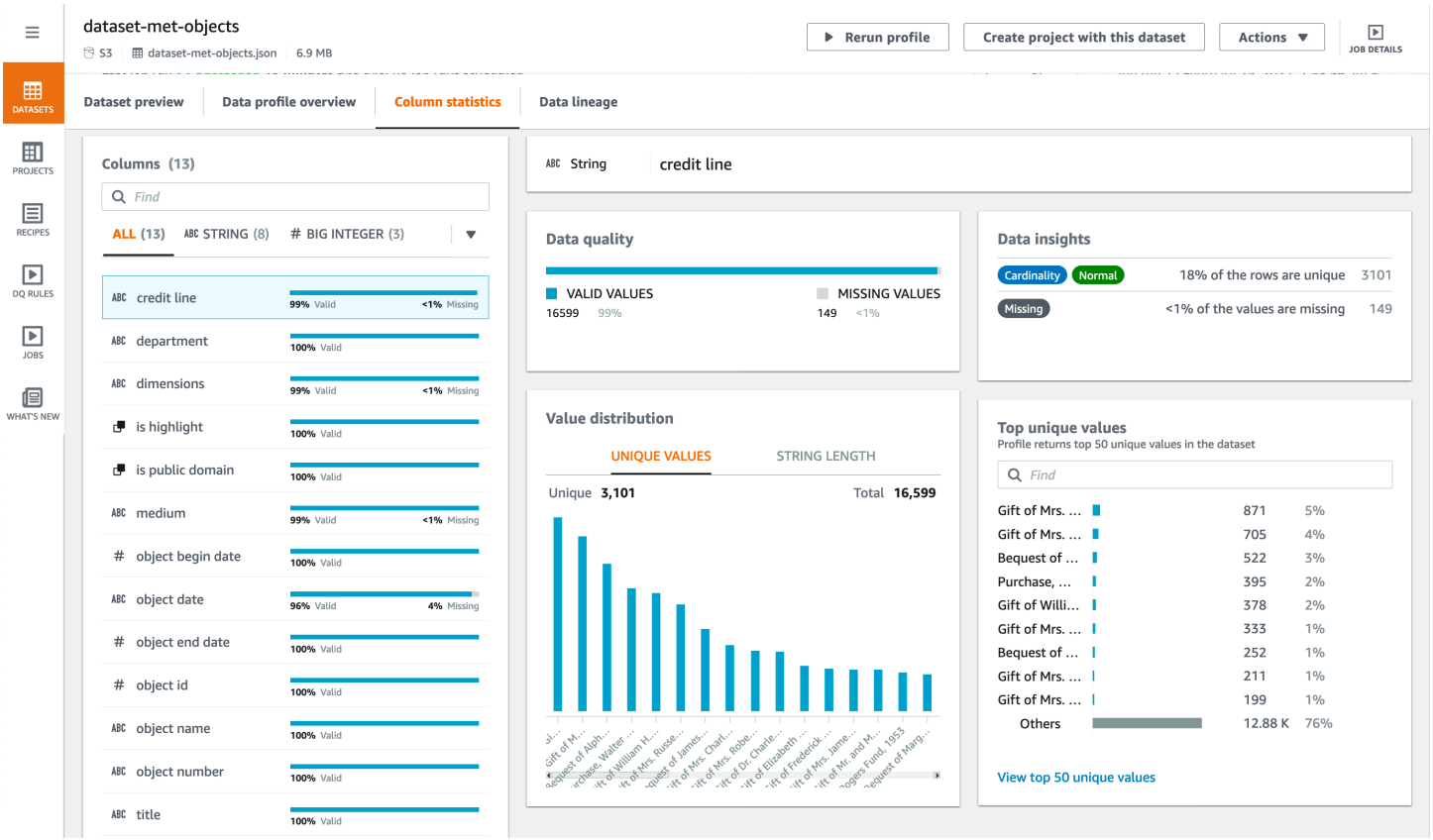


Visão geral do perfil de dados — Nessa guia, você pode encontrar um perfil gráfico de dados estatísticos e volumétricos do seu conjunto de dados, conforme mostrado a seguir.

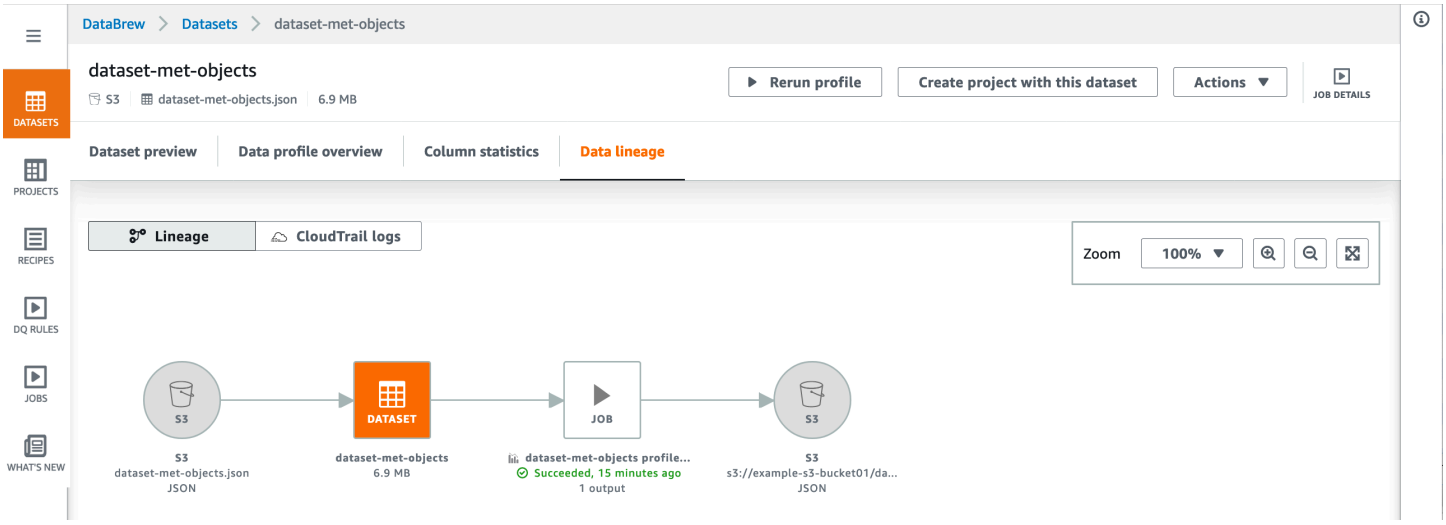


> **Note**
>
> Para criar um perfil de dados, execute um trabalho DataBrew de perfil no seu conjunto de dados. Para obter informações sobre como fazer isso, consulte Etapa 5: criar um perfil de dados.

Estatísticas da coluna — Nessa guia, você pode encontrar estatísticas detalhadas sobre cada coluna em seu conjunto de dados, conforme mostrado a seguir.



Linhagem de dados — Essa guia mostra uma representação gráfica de como seu conjunto de dados foi criado e como ele é usado DataBrew, conforme mostrado a seguir.



Tópicos

- Excluir um conjunto de dados

Excluir um conjunto de dados

Se você não precisar mais de um conjunto de dados, poderá excluí-lo. A exclusão de um conjunto de dados não afeta a fonte de dados subjacente de forma alguma. Ele simplesmente remove as informações DataBrew usadas para acessar a fonte de dados.

Você não pode excluir um conjunto de dados se outros DataBrew recursos dependerem dele. Por exemplo, se você tem atualmente um DataBrew projeto que usa o conjunto de dados, exclua o projeto antes de excluir o conjunto de dados.

Para excluir um conjunto de dados, escolha Conjunto de dados no painel de navegação. Escolha o conjunto de dados que você deseja excluir e, em Ações, escolha Excluir.

Conectando-se aos seus dados

Para obter mais informações sobre como se conectar às seguintes fontes de dados, escolha a seção que se aplica a você.

- AWS Glue Data Catalog— Você pode usar o Catálogo de Dados para definir referências a objetos de dados armazenados na AWS nuvem, incluindo os seguintes serviços:
  - banco de dados de origem
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS para MySQL
  - Amazon RDS para PostgreSQL

  DataBrew reconhece todas as permissões do Lake Formation que foram aplicadas aos recursos do Catálogo de Dados, portanto, DataBrew os usuários só podem acessar esses recursos se estiverem autorizados.

  Para criar um conjunto de dados, você especifica um nome de banco de dados do Catálogo de Dados e um nome de tabela. DataBrew cuida dos outros detalhes da conexão.
- AWS Troca de dados — Você pode escolher entre centenas de fontes de dados de terceiros que estão disponíveis no AWS Data Exchange. Ao assinar essas fontes de dados, você sempre tem a versão mais atualizada dos dados.

  Para criar um conjunto de dados, você especifica o nome de um produto de dados do Data Exchange que você está inscrito ou autorizado a usar.

- Conexões do driver JDBC — Você pode criar um conjunto de dados conectando-se DataBrew a uma JDBC-compatible fonte de dados. DataBrew suporta a conexão com as seguintes fontes por meio do JDBC:
  - banco de dados de origem
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Snowflake

Tópicos

Usando drivers com AWS Glue DataBrew

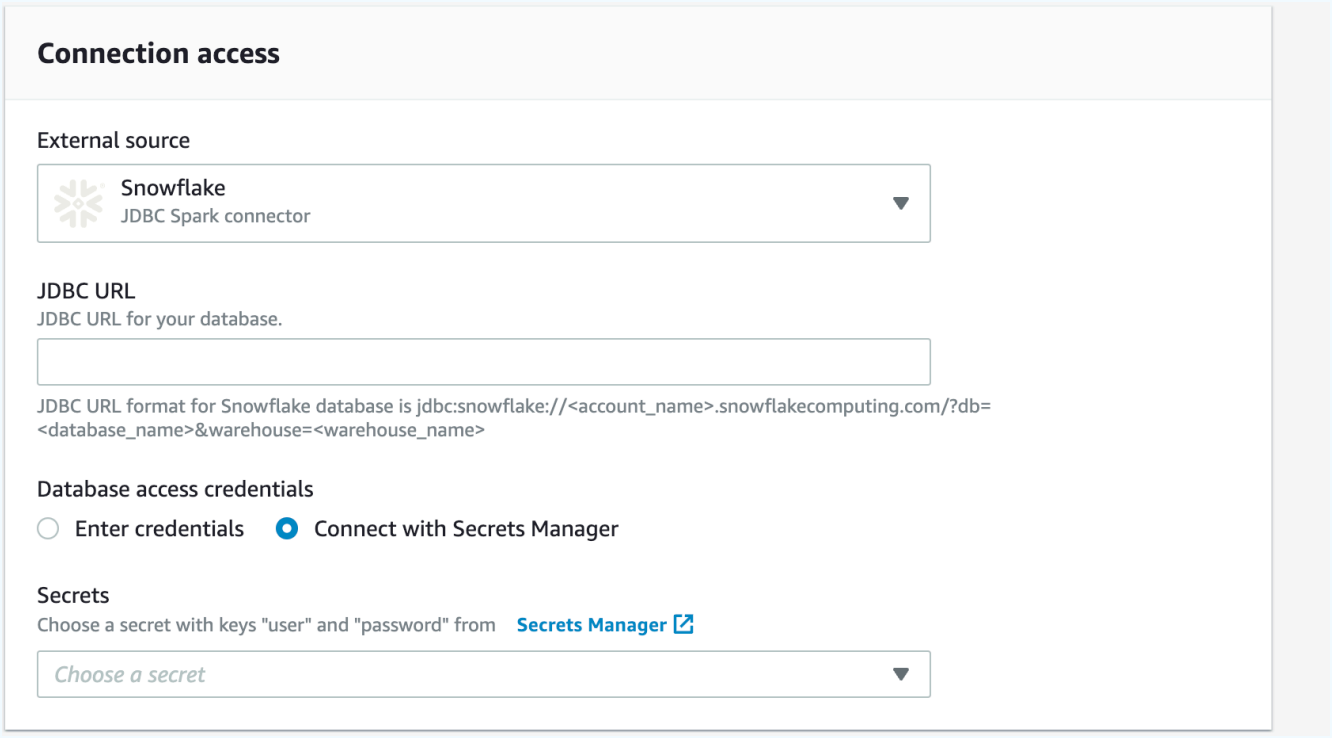
Um driver de banco de dados é um arquivo ou URL que implementa um protocolo de conexão de banco de dados, por exemplo, Java Database Connectivity (JDBC). O driver funciona como um adaptador ou tradutor entre um sistema de gerenciamento de banco de dados específico (DBMS) e outro sistema.

Nesse caso, ele permite AWS Glue DataBrew conectar-se aos seus dados. Em seguida, você pode acessar um objeto de banco de dados, como uma tabela ou exibição, a partir de uma fonte de dados compatível. A fonte de dados que você está usando pode ser chamada de banco de dados, data warehouse ou outra coisa. No entanto, para fins desta documentação, nos referimos a todos os provedores de dados como fontes de dados ou conexões.

Para usar um driver JDBC ou arquivo jar, baixe o arquivo ou os arquivos necessários e coloque-os em um bucket do S3. A função do IAM que você usa para acessar os dados precisa ter permissões de leitura para os dois arquivos do driver.

> **Note**
>
> With AWS Glue4.0, a conexão com o Snowflake como fonte de dados é suportada nativamente. Você não precisa fornecer `jar` arquivos personalizados. Em AWS Glue DataBrew, escolha Snowflake como conexão de origem externa e forneça a URL da

sua instância do Snowflake. O URL usará um nome de host no formulário `https://account_identifier.snowflakecomputing.com`.
Forneça as credenciais de acesso aos dados, o nome do banco de dados do Snowflake e o nome do esquema do Snowflake. Além disso, se o usuário do Snowflake não tiver um conjunto de depósito padrão, você precisará fornecer um nome de depósito.
As conexões do Snowflake usam um AWS Secrets Manager segredo para fornecer informações de credenciais. Seu projeto e suas funções profissionais devem ter permissão para ler esse segredo.



Para usar drivers com DataBrew

1. Descubra em qual versão da sua fonte de dados você está usando o método fornecido pelo produto.
2. Encontre a versão mais recente dos conectores e do driver necessário. Você pode localizar essas informações no site do provedor de dados.
3. Baixe a versão necessária dos arquivos JDBC. Normalmente, eles são armazenados como arquivos Java Archives (.JAR).
4. Faça o upload dos drivers do console para o bucket do S3 ou forneça o caminho do S3 para seus arquivos.JAR.

5. Insira os detalhes básicos da conexão, por exemplo, classe, instância e assim por diante.
6. Insira qualquer informação de configuração adicional que sua fonte de dados precise, por exemplo, informações de nuvem privada virtual (VPC).

Drivers JDBC compatíveis

Produto	Versão do compatível	Instruções e downloads do driver	Consultas SQL suportadas
Microsoft SQL Server	v6.x ou superior	Driver Microsoft JDBC para SQL Server	Não compatível
MySQL	v5.1 ou superior	Conectores MySQL	Não compatível
Oracle	v11.2 ou superior	Downloads do Oracle JDBC	Não compatível
PostgreSQL	v4.2.x ou superior	Controlador JDBC PostgreSQL	Não compatível
banco de dados de origem	v4.1 ou superior	Conectando-se ao Amazon Redshift com o JDBC	Compatível
Snowflake	Para ver sua	Para se conectar ao Snowflake, você precisa das duas opções a seguir:	Compatível

Produto	Versão do compatível	Instruções e downloads do driver	Consultas SQL suportadas
	versão do Snowflake, use CURRENT_VERSION conforme descrito na documentação do Snowflake.	• Controlador JDBC Snowflake • Conector Snowflake para Spark	

Para se conectar a bancos de dados ou armazéns de dados que exigem uma versão do driver diferente da que é suportada DataBrew nativamente, você pode fornecer um driver JDBC de sua escolha. O driver deve ser compatível com JDK 8 ou Java 8. Para obter instruções sobre como encontrar a versão mais recente do driver para seu banco de dados, consulteUsando drivers com AWS Glue DataBrew.

Conectando-se aos dados em um arquivo de texto com DataBrew

Você pode configurar as seguintes opções de formato para os arquivos de entrada que oferecem DataBrew suporte a:

- Comma-separated arquivos de valor (CSV)
  - Delimitadores

    O delimitador padrão é uma vírgula para arquivos.csv. Se seu arquivo usa um delimitador diferente, escolha o delimitador para o delimitador CSV na seção Configurações adicionais ao criar seu conjunto de dados. Os seguintes delimitadores são compatíveis com arquivos.csv:
    - Vírgula (,)

- Cólon (:)
- Semi-colon (;)
- Barra vertical (|)
- Tabulação (\t)
- Curvo circunflexo (^)
- Barra invertida (\)
- Space

- Valores do cabeçalho da coluna

  Seu arquivo CSV pode incluir uma linha de cabeçalho como a primeira linha do arquivo. Caso contrário, DataBrew cria uma linha de cabeçalho para você.

  - Se o arquivo CSV incluir uma linha de cabeçalho, escolha Tratar a primeira linha como cabeçalho. Se você fizer isso, a primeira linha do seu arquivo CSV será tratada como contendo os valores do cabeçalho da coluna.
  - Se o arquivo CSV não incluir uma linha de cabeçalho, escolha Adicionar cabeçalho padrão. Se você fizer isso, DataBrew cria uma linha de cabeçalho para o arquivo e não trata sua primeira linha de dados como contendo valores de cabeçalho. Os cabeçalhos DataBrew criados consistem em um sublinhado e um número para cada coluna no arquivo, no formato`Column_1`, `Column_2``Column_3`, e assim por diante.

- Arquivos JSON

  DataBrew suporta dois formatos para arquivos JSON, linhas JSON e documento JSON. Os arquivos JSON Lines contêm uma linha por linha. Nos arquivos de documentos JSON, todas as linhas estão contidas em uma única estrutura JSON ou em uma matriz. Você pode especificar seu tipo de arquivo JSON na seção Configurações adicionais ao criar um conjunto de dados JSON. O formato padrão é Linhas JSON.

- Arquivos do Excel

  O seguinte se aplica às planilhas do Excel em DataBrew:

  - Carregando planilhas do Excel

    Por padrão, DataBrew carrega a primeira planilha em seu arquivo Excel. No entanto, você pode especificar um número ou nome de folha diferente na seção Configurações adicionais ao criar um conjunto de dados do Excel.

  - Valores do cabeçalho da coluna

Suas planilhas do Excel podem incluir uma linha de cabeçalho como a primeira linha do arquivo, mas se não incluírem, DataBrew criarão uma linha de cabeçalho para você.

- Se suas planilhas do Excel incluírem uma linha de cabeçalho, escolha Tratar a primeira linha como cabeçalho. Se você fizer isso, a primeira linha das planilhas do Excel será tratada como contendo os valores do cabeçalho da coluna.
- Se o arquivo do Excel não incluir uma linha de cabeçalho, escolha Adicionar cabeçalho padrão. Ao fazer isso, você especifica que DataBrew deve criar uma linha de cabeçalho para o arquivo e não tratar sua primeira linha de dados como contendo valores de cabeçalho. Os cabeçalhos DataBrew criados consistem em um sublinhado e um número para cada coluna no arquivo, no formato`Column_1`, `Column_2Column_3`, e assim por diante.

Conectando dados em vários arquivos no Amazon S3

Com o DataBrew console, você pode navegar pelos buckets e pastas do Amazon S3 e escolher um arquivo para seu conjunto de dados. No entanto, um conjunto de dados não precisa ser limitado a um arquivo.

Suponha que você tenha um bucket do S3 chamado `my-databrew-bucket` que contém uma pasta chamada`databrew-input`. Nessa pasta, suponha que você tenha vários arquivos JSON, todos com o mesmo formato e extensão de `.json` arquivo. No console, você pode especificar uma URL de origem de`s3://my-databrew-bucket/databrew-input/`. No DataBrew console, você pode então escolher essa pasta. Seu conjunto de dados consiste em todos os arquivos JSON dessa pasta.

DataBrew pode processar todos os arquivos em uma pasta do S3, mas somente se as seguintes condições forem verdadeiras:

- Todos os arquivos na pasta têm o mesmo formato.
- Todos os arquivos na pasta têm a mesma extensão de arquivo.

Para obter mais informações sobre extensões e formatos de arquivo compatíveis, consulte[DataBrew input formats](#).

Esquemas ao usar vários arquivos como conjunto de dados

Ao usar vários arquivos como DataBrew conjunto de dados, os esquemas precisam ser os mesmos em todos os arquivos. Caso contrário, o Project Workspace tentará escolher automaticamente um dos esquemas dos vários arquivos e tentará adequar o restante dos arquivos do conjunto de dados a esse esquema. Esse comportamento faz com que a exibição exibida durante o Project Workspace seja irregular e, como resultado, a saída do trabalho também será irregular.

Se seus arquivos precisarem ter esquemas diferentes, você precisará criar vários conjuntos de dados e perfilá-los separadamente.

Usando caminhos parametrizados para o Amazon S3

Em alguns casos, talvez você queira criar um conjunto de dados com arquivos que sigam uma determinada convenção de nomenclatura ou um conjunto de dados que possa abranger várias pastas do Amazon S3. Ou talvez você queira reutilizar o mesmo conjunto de dados para dados estruturados de forma idêntica que são gerados periodicamente em um local do S3 com um caminho que depende de determinados parâmetros. Um exemplo é um caminho com o nome da data de produção de dados.

DataBrew suporta essa abordagem com caminhos S3 parametrizados. Um caminho parametrizado é uma URL do Amazon S3 contendo expressões regulares ou parâmetros de caminho personalizados, ou ambos.

Definindo um conjunto de dados com um caminho do S3 usando expressões regulares

Expressões regulares no caminho podem ser úteis para combinar vários arquivos de uma ou mais pastas e, ao mesmo tempo, filtrar arquivos não relacionados nessas pastas.

Aqui estão alguns exemplos:

- Defina um conjunto de dados incluindo todos os arquivos JSON de uma pasta cujo nome começa com. `invoice`
- Defina um conjunto de dados incluindo todos os arquivos em pastas com `2020` seus nomes.

Você pode implementar esse tipo de abordagem usando expressões regulares em um caminho S3 do conjunto de dados. Essas expressões regulares podem substituir qualquer substring na chave do URL do S3 (mas não o nome do bucket).

Como exemplo de uma chave em um URL do S3, veja o seguinte. Aqui `my-bucket` está o nome do bucket, Leste dos EUA (Ohio) é a AWS região e `puppy.png` é o nome da chave.

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

Em um caminho S3 parametrizado, todos os caracteres entre dois colchetes angulares (<e>) são tratados como expressões regulares. Dois exemplos são os seguintes:

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json`corresponde a todos os arquivos nomeados`data.json`, em todas as subpastas `databrew-input` cujos nomes começam com`invoice`.
- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/`corresponde a todos os arquivos em pastas com `2020` seus nomes.

Nesses exemplos, `.*` corresponde a zero ou mais caracteres.

> **Note**
>
> Você só pode usar expressões regulares na parte principal do caminho do S3 — a parte que vem depois do nome do bucket. Portanto, `s3://my-databrew-bucket/<.*>-input/` é válido, mas `s3://my-<.*>-bucket/<.*>-input/` não é.

Recomendamos que você teste suas expressões regulares para garantir que elas correspondam somente aos URLs do S3 que você deseja, e não aos que você não deseja.

Aqui estão alguns outros exemplos de expressões regulares:

- `<\d{2}>`corresponde a uma string que consiste em exatamente dois dígitos consecutivos, por exemplo `07` ou`03`, mas não`1a2`.
- `<[a-z]+.*>`corresponde a uma string que começa com uma ou mais letras latinas minúsculas e tem zero ou mais caracteres depois dela. Um exemplo é `a3abc/def`, ou`a-z`, mas não`A2`.
- `<[^/]+>`corresponde a uma string que contém qualquer caractere, exceto uma barra (/). Em uma URL do S3, barras são usadas para separar pastas no caminho.
- `<.*=.*>`corresponde a uma string que contém um sinal de igual (=), por exemplo, ou `month=02` `abc/day=2=10`, mas não. `test`
- `<\d.*\d>`corresponde a uma string que começa e termina com um dígito e pode ter qualquer outro caractere entre os dígitos, por exemplo`1abc2`, ou `01-02-032020/Jul/21`, mas não. `123a`

Definindo um conjunto de dados com um caminho do S3 usando parâmetros personalizados

Definir um conjunto de dados parametrizado usando parâmetros personalizados oferece vantagens em relação ao uso de expressões regulares quando você pode querer fornecer parâmetros para um local do S3:

- Você pode obter os mesmos resultados de uma expressão regular, sem precisar conhecer a sintaxe das expressões regulares. Você pode definir parâmetros usando termos familiares como “começa com” e “contém”.
- Ao definir um conjunto de dados dinâmico usando parâmetros no caminho, você pode incluir um intervalo de tempo em sua definição, como “mês passado” ou “últimas 24 horas”. Dessa forma, sua definição de conjunto de dados será usada posteriormente com novos dados recebidos.

Aqui estão alguns exemplos de quando você pode querer usar conjuntos de dados dinâmicos:

- Para conectar vários arquivos particionados pela data da última atualização ou outros atributos significativos em um único conjunto de dados. Em seguida, você pode capturar esses atributos de partição como colunas adicionais em um conjunto de dados.
- Para restringir arquivos em um conjunto de dados a locais do S3 que atendam a determinadas condições. Por exemplo, suponha que seu caminho do S3 contenha pastas baseadas em datas, como. `folder/2021/04/01/` Nesse caso, você pode parametrizar a data e restringi-la a um determinado intervalo, como “entre 01 de março de 2021 e 01 de abril de 2021" ou “Semana passada”.

Para definir um caminho usando parâmetros, defina os parâmetros e adicione-os ao seu caminho usando o seguinte formato:

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

> **Note**
>
> Assim como acontece com as expressões regulares em um caminho do S3, você só pode usar parâmetros na parte principal do caminho — a parte que vem depois do nome do bucket.

Dois campos são obrigatórios em uma definição de parâmetro, nome e tipo. O tipo pode ser Cadeia de caracteres, Número ou Data. Os parâmetros do tipo Data devem ter uma definição do formato da data para que DataBrew possam interpretar e comparar corretamente os valores da data. Opcionalmente, você pode definir condições de correspondência para um parâmetro. Você também pode optar por adicionar valores correspondentes de um parâmetro como uma coluna ao seu conjunto de dados quando ele está sendo carregado por um DataBrew trabalho ou sessão interativa.

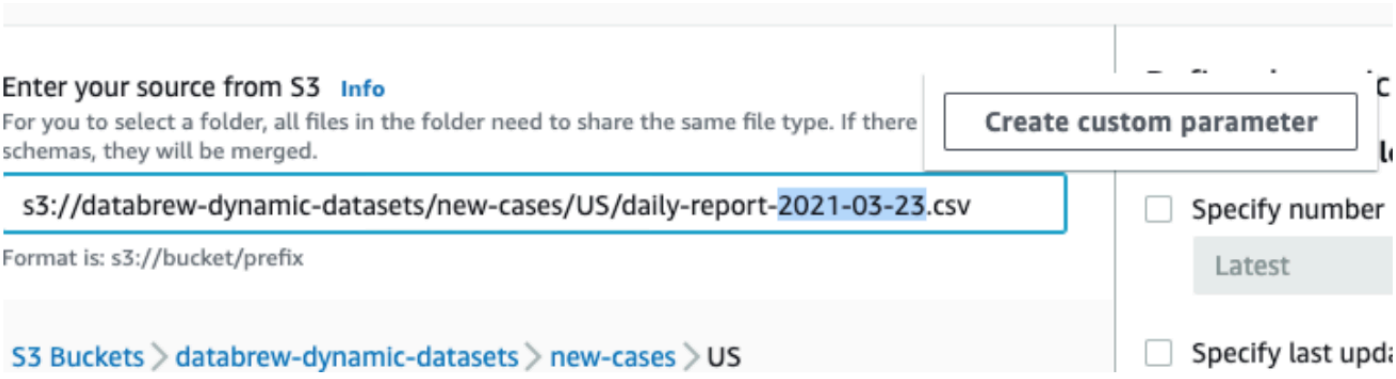
Exemplo

Vamos considerar um exemplo de definição de um conjunto de dados dinâmico usando parâmetros no DataBrew console. Neste exemplo, suponha que os dados de entrada sejam gravados regularmente em um bucket do S3 usando locais como estes:

- `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`

Há duas partes dinâmicas aqui: um código de país, como EUA, e uma data no nome do arquivo, como 2021-03-30. Aqui, você pode aplicar a mesma receita de limpeza para todos os arquivos. Digamos que você queira realizar seu trabalho de limpeza diariamente. Veja a seguir como você pode definir um caminho parametrizado para esse cenário:

1. Navegue até um arquivo específico.
2. Em seguida, selecione uma peça variável, como uma data, e substitua-a por um parâmetro. Nesse caso, substitua uma data.



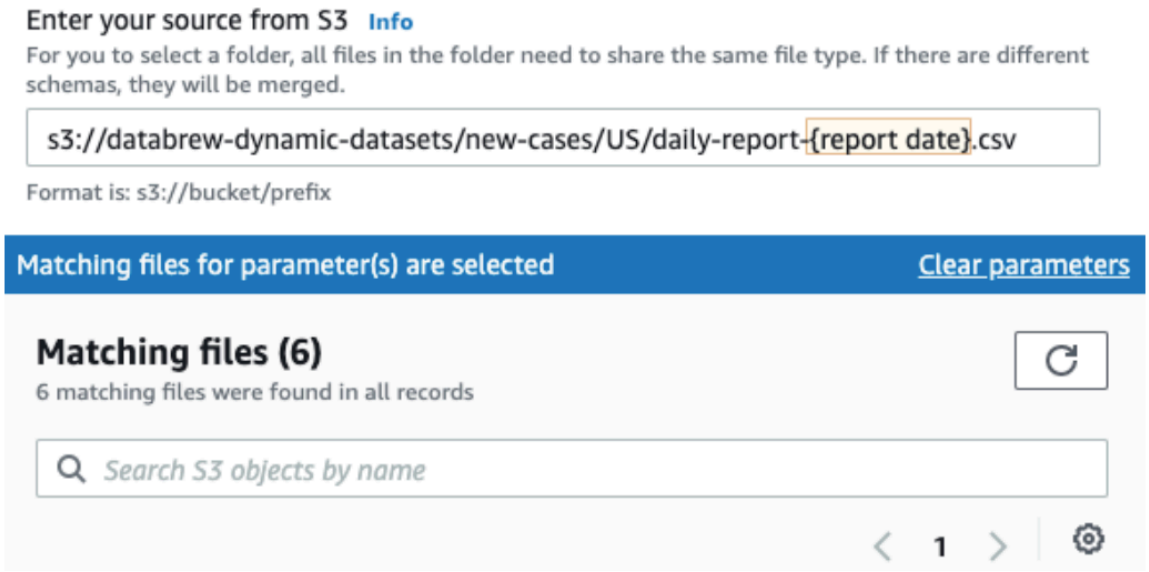
3. Abra o menu de contexto (clique com o botão direito do mouse) para Criar parâmetro personalizado e defina as propriedades para ele:

- Nome: data do relatório
- Tipo: data
- Formato de data: aaaa- MM-dd (selecionado a partir dos formatos predefinidos)
- Condições (intervalo de tempo): últimas 24 horas
- Adicionar como coluna: verdadeiro (verificado)

Mantenha os outros campos em seus valores padrão.

4. Escolha Criar.

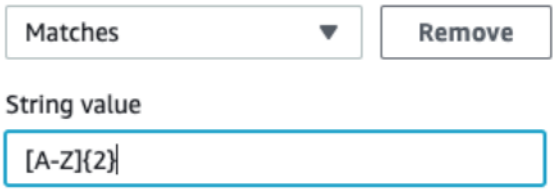
Depois disso, você verá o caminho atualizado, como na captura de tela a seguir.



Agora você pode fazer o mesmo com o código do país e parametrizá-lo da seguinte forma:

- Nome: código do país
- Tipo: string
- Adicionar como coluna: verdadeiro (verificado)

Você não precisa especificar condições se todos os valores forem relevantes. Na `new-cases` pasta, por exemplo, só temos subpastas com códigos de país, então não há necessidade de condições. Se você tiver outras pastas para excluir, poderá usar a condição a seguir.



Essa abordagem limita as subpastas de novos casos a conter dois caracteres latinos maiúsculos.

Após essa parametrização, você tem apenas arquivos correspondentes em nosso conjunto de dados e pode escolher Criar conjunto de dados.

> **Note**
>
> Quando você usa intervalos de tempo relativos em condições, os intervalos de tempo são avaliados quando o conjunto de dados é carregado. Isso é verdade se forem intervalos de tempo predefinidos, como “Últimas 24 horas”, ou intervalos de tempo personalizados, como “5 dias atrás”. Essa abordagem de avaliação se aplica se o conjunto de dados for carregado durante a inicialização de uma sessão interativa ou durante o início de um trabalho.

Depois de escolher Criar conjunto de dados, seu conjunto de dados dinâmico estará pronto para uso. Por exemplo, você pode usá-lo primeiro para criar um projeto e definir uma receita de limpeza usando uma DataBrew sessão interativa. Em seguida, você pode criar um trabalho programado para ser executado diariamente. Esse trabalho pode aplicar a receita de limpeza aos arquivos do conjunto de dados que atendam às condições de seus parâmetros no momento em que o trabalho é iniciado.

Condições suportadas para conjuntos de dados dinâmicos

Você pode usar condições para filtrar arquivos S3 correspondentes usando parâmetros ou o atributo de data da última modificação.

A seguir, você encontrará listas de condições suportadas para cada tipo de parâmetro.

Condições usadas com parâmetros String

Nome no DataBrew SDK	Sinônimos de SDK	Nome no DataBrew console	Description
é	equação, ==	É exatamente	O valor do parâmetro é igual ao valor

Nome no DataBrew SDK	Sinônimos de SDK	Nome no DataBrew console	Description
			fornecido na condição.
não é	não eq,! =	Não é	O valor do parâmetro não é o mesmo que o valor fornecido na condição.
contém		Contém	O valor da string do parâmetro contém o valor fornecido na condição.
não contém		Não contém	O valor da string do parâmetro não contém o valor fornecido na condição.
começa_com		Inicia com	O valor da string do parâmetro começa com o valor fornecido na condição.
não começa com		Não começa com	O valor da string do parâmetro não começa com o valor fornecido na condição.
ends_with		Termina com	O valor da string do parâmetro termina com o valor fornecido na condição.

Nome no DataBrew SDK	Sinônimos de SDK	Nome no DataBrew console	Description
não termina com		Não termina com	O valor da string do parâmetro não termina com o valor fornecido na condição.
matches		Correspondências	O valor do parâmetro corresponde à expressão regular fornecida na condição.
não corresponde		Não coincide	O valor do parâmetro não correspon de à expressão regular fornecida na condição.

> **Note**
>
> Todas as condições dos parâmetros String usam comparação com distinção entre maiúsculas e minúsculas. Se você não tiver certeza sobre o caso usado em um caminho do S3, você pode usar a condição “matches” com um valor de expressão regular que começa com `(?i)`. Fazer isso resulta em uma comparação sem distinção entre maiúsculas e minúsculas.
>
> Por exemplo, suponha que você queira que seu parâmetro de string comece com `abc`, mas `Abc` também seja possível. `ABC` Nesse caso, você pode usar a condição “matches” `(?i)^abc` como valor da condição.

Condições usadas com parâmetros numéricos

Nome no DataBrew SDK	Sinônimos de SDK	Nome no DataBrew console	Description
é	equação, ==	É exatamente	O valor do parâmetro é igual ao valor fornecido na condição.
não é	não eq,! =	Não é	O valor do parâmetro não é o mesmo que o valor fornecido na condição.
menos_que	lt, <	Menor que	O valor numérico do parâmetro é menor que o valor fornecido na condição.
menos_que_igual	tarde, <=	Menor ou igual a	O valor numérico do parâmetro é menor ou igual ao valor fornecido na condição.
maior_que	gt, >	Maior que	O valor numérico do parâmetro é maior que o valor fornecido na condição.
maior_que_igual	obter, =>	Maior ou igual a	O valor numérico do parâmetro é maior ou igual ao valor fornecido na condição.

Condições usadas com parâmetros de data

Nome no DataBrew SDK	Nome no DataBrew console	Formato do valor da condição (SDK)	Description
after	Início	Formato de data ISO 8601, como ou `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	O valor do parâmetro de data é posterior à data fornecida na condição.
antes	Fim	Formato de data ISO 8601, como ou `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	O valor do parâmetro de data é anterior à data fornecida na condição.
relativo_depois	Início (relativo)	Número positivo ou negativo de unidades de tempo, como `-48h` ou`+7d`.	O valor do parâmetro de data é posterior à data relativa fornecida na condição. As datas relativas são avaliadas quando o conjunto de dados é carregado, quando uma sessão interativa é inicializada ou quando um trabalho associado é iniciado. Esse é o momento chamado de “agora” nos exemplos.
relativo_antes	Fim (relativo)	Número positivo ou negativo de unidades	O valor do parâmetro de data é anterior à

Nome no DataBrew SDK	Nome no DataBrew console	Formato do valor da condição (SDK)	Description
		de tempo, como `-48h` ou`+7d`.	data relativa fornecida na condição. As datas relativas são avaliadas quando o conjunto de dados é carregado, quando uma sessão interativ a é inicializada ou quando um trabalho associado é iniciado. Esse é o momento chamado de “agora” nos exemplos.

Se você usa o SDK, forneça datas relativas no seguinte formato:`±{number_of_time_units}{time_unit}`. Você pode usar essas unidades de tempo:

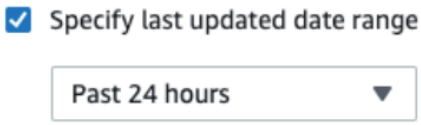
- -1h (1 hora atrás)
- +2d (a partir de agora 2 dias)
- -120m (120 minutos atrás)
- 5000s (5.000 segundos a partir de agora)
- -3w (3 semanas atrás)
- +4M (daqui a 4 meses)
- -1y (1 ano atrás)

As datas relativas são avaliadas quando o conjunto de dados é carregado, quando uma sessão interativa é inicializada ou quando um trabalho associado é iniciado. Esse é o momento chamado de “agora” nos exemplos anteriores.

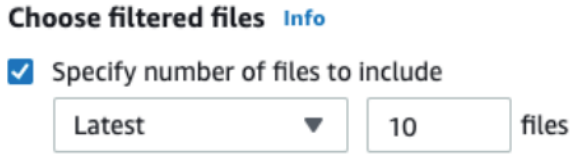
Definindo configurações para conjuntos de dados dinâmicos

Além de fornecer um caminho S3 parametrizado, você pode definir outras configurações para conjuntos de dados com vários arquivos. Essas configurações filtram os arquivos do S3 pela data da última modificação e limitam o número de arquivos.

Semelhante à configuração de um parâmetro de data em um caminho, você pode definir um intervalo de tempo em que os arquivos correspondentes foram atualizados e incluir somente esses arquivos em seu conjunto de dados. Você pode definir esses intervalos usando datas absolutas, como “30 de março de 2021", ou intervalos relativos, como “Últimas 24 horas”.



Para limitar o número de arquivos correspondentes, selecione um número de arquivos maior que 0 e se você deseja os arquivos correspondentes mais recentes ou os mais antigos.



Tipos de dados

Os dados de cada coluna do seu conjunto de dados são convertidos em um dos seguintes tipos de dados:

- byte — números inteiros assinados de 1 byte. O intervalo de números é de -128 a 127.
- curto — números inteiros assinados de 2 bytes. O intervalo de números é de -32768 a 32767.
- inteiro — números inteiros assinados de 4 bytes. O intervalo de números é de -2147483648 a 2147483647.
- longo — números inteiros assinados de 8 bytes. O intervalo de números é de -9223372036854775808 a 9223372036854775807.
- float — números de ponto flutuante de precisão única de 4 bytes.
- double — números de ponto flutuante de precisão dupla de 8 bytes.
- decimal — Números decimais assinados com até 38 dígitos no total e 18 dígitos após o ponto decimal.

- string — Valores da cadeia de caracteres.
- booleano — O tipo booleano tem um dos dois valores possíveis: `true` e `false` ou `yes` e `no`.
- timestamp — Valores que incluem os campos ano, mês, dia, hora, minuto e segundo.
- data — Valores que compreendem os campos ano, mês e dia.

Tipos de dados avançados

Os tipos de dados avançados são tipos de dados que são DataBrew detectados em uma coluna de string em um projeto e, portanto, não fazem parte de um conjunto de dados. Para obter informações sobre tipos de dados avançados, consulte Tipos de dados avançados.

Tipos de dados avançados

Os tipos de dados avançados são tipos de dados que são DataBrew detectados em uma coluna de string em um projeto por meio da correspondência de padrões. Quando você clica em uma coluna de sequência de caracteres, a coluna é marcada como o tipo de dados avançado correspondente se 50% ou mais dos valores na coluna atenderem aos critérios desse tipo de dados.

Os tipos de dados DataBrew que posso detectar são:

- Date/timestamp
- SSN
- Número de telefone
- E-mail
- Cartão de crédito
- Gender
- IP address (endereço de IP)
- URL
- CEP
- País
- Moeda
- Estado
- Cidade

Você pode usar as seguintes transformações para trabalhar com tipos de dados avançados:

- GET_ADVANCED_DATATYPE: dada uma coluna de sequência de caracteres, identifica o tipo de dados avançado da coluna, se houver.
- EXTRAIR DETALHES AVANÇADOS DO TIPO DE DADOS: extrai detalhes de um tipo de dados avançado.
- FILTRO_DE_TIPO DE DADOS AVANÇADO: filtra uma coluna de origem atual com base na detecção avançada do tipo de dados.
- ADVANCED_DATATYPE_FLAG: cria uma nova coluna de sinalização com base nos valores da coluna de origem atual.

Validando a qualidade dos dados em AWS Glue DataBrew

Para garantir a qualidade dos seus conjuntos de dados, você pode definir uma lista de regras de qualidade de dados em um conjunto de regras. Um conjunto de regras é um conjunto de regras que compara diferentes métricas de dados com valores esperados. Se algum dos critérios de uma regra não for atendido, o conjunto de regras como um todo falhará na validação. Em seguida, você pode inspecionar os resultados individuais de cada regra. Para qualquer regra que cause uma falha na validação, você pode fazer as correções necessárias e revalidar.

Exemplos de regras incluem o seguinte:

- O valor na coluna `"APY"` está entre 0 e 100
- O número de valores faltantes na coluna `group_name` não excede 5%

Você pode definir cada regra para uma coluna individual ou aplicá-la de forma independente a várias colunas selecionadas, por exemplo:

- O valor máximo não excede 100 para colunas`"rate"`,`"pay"`,`"increase"`.

Uma regra pode consistir em várias verificações simples. Você pode definir se todas elas devem ser verdadeiras ou alguma, por exemplo:

- O valor na coluna `"ProductId"` deve começar com `"asin-"` E o comprimento do valor na coluna `"ProductId"` é 32.

Você pode verificar as regras em relação a valores agregados`max`, como`min`, ou `number of duplicate values` onde há apenas um valor sendo comparado, ou valores não agregados em cada linha de uma coluna. No último caso, você também pode definir um limite de “passagem”, como`value in columnA > value in columnB for at least 95% of rows`.

Assim como acontece com as informações de perfil, você pode definir regras de qualidade de dados em nível de coluna somente para colunas de tipos simples, como cadeias de caracteres e números. Você não pode definir regras de qualidade de dados para colunas de tipos complexos, como matrizes ou estruturas. Para obter mais detalhes sobre como trabalhar com informações de perfil, consulteCriando e trabalhando com AWS Glue DataBrew empregos de perfil.

Validando regras de qualidade de dados

Depois que um conjunto de regras for definido, você poderá adicioná-lo a um trabalho de perfil para validação. Você pode definir mais de um conjunto de regras para um conjunto de dados.

Por exemplo, um conjunto de regras pode conter regras com critérios minimamente aceitáveis. Uma falha na validação desse conjunto de regras pode significar que os dados não são aceitáveis para uso posterior. Um exemplo são valores ausentes nas principais colunas de um conjunto de dados usado para treinamento de aprendizado de máquina. Você pode usar um segundo conjunto de regras com regras mais rígidas para verificar se o conjunto de dados tem uma qualidade tão boa que não é necessária nenhuma limpeza.

Você pode aplicar um ou mais conjuntos de regras definidos para um determinado conjunto de dados em uma configuração de trabalho de perfil. Quando a tarefa do perfil é executada, ela produz um relatório de validação além do perfil de dados. O relatório de validação está disponível no mesmo local dos dados do seu perfil. Assim como acontece com as informações do perfil, você pode explorar os resultados no DataBrew console. Na visualização de detalhes do conjunto de dados, escolha a guia Qualidade de dados para ver os resultados. Para obter mais detalhes sobre como trabalhar com informações de perfil, consulteCriando e trabalhando com AWS Glue DataBrew empregos de perfil.

Atuando nos resultados da validação

Quando um trabalho DataBrew de perfil é concluído, DataBrew envia um CloudWatch evento da Amazon com os detalhes desse trabalho executado. Se você também configurou seu trabalho para validar as regras de qualidade de dados, DataBrew envia um evento para cada conjunto de regras validado. O evento contém seu resultado (`SUCCEEDED`,`FAILED`, ou`ERROR`) e um link para o relatório detalhado de validação da qualidade de dados. Em seguida, você pode automatizar outras ações invocando a próxima ação, dependendo do status da validação. Para obter mais informações sobre como conectar eventos a ações de destino, como notificação do Amazon SNS, invocações de AWS Lambda funções e outras, consulte Introdução à Amazon. EventBridge

Veja a seguir um exemplo de um evento de resultado de DataBrew validação:

```
{
  "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"
  }
}
```

Você pode usar atributos de eventos`detail-type`, como, `source` e propriedades aninhadas do `detail` atributo para criar padrões de eventos no Amazon Eventbridge. Por exemplo, um padrão de evento que correspondesse a todas as validações com falha de qualquer DataBrew trabalho teria a seguinte aparência:

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

Para obter um exemplo de como criar um conjunto de regras e validar suas regras, consulte. Criação de um conjunto de regras com regras de qualidade de dados Para obter mais informações sobre como trabalhar com CloudWatch eventos em DataBrew, consulte Automatização com eventos DataBrew CloudWatch

Criação de um conjunto de regras com regras de qualidade de dados

No procedimento a seguir, você pode encontrar um exemplo de como criar um conjunto de regras e aplicá-lo a um conjunto de dados. Um conjunto de regras é um conjunto de regras que compara

diferentes métricas de dados com valores esperados. Em seguida, você pode usar esse conjunto de regras em um trabalho de perfil para validar as regras de qualidade de dados que ele inclui.

Para criar um exemplo de conjunto de regras com regras de qualidade de dados

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console em https://console.aws.amazon.com/databrew/.
2. Escolha DQ RULES no painel de navegação e escolha Criar conjunto de regras de qualidade de dados.
3. Insira um nome para seu conjunto de regras. Opcionalmente, insira uma descrição para seu conjunto de regras.
4. Em Conjunto de dados associado, escolha um conjunto de dados para associar ao conjunto de regras.

   Depois de selecionar um conjunto de dados, você pode visualizar o painel de visualização do conjunto de dados à direita.
5. Use a visualização no painel de visualização do conjunto de dados para explorar os valores e o esquema do conjunto de dados ao determinar as regras de qualidade de dados a serem criadas. A visualização prévia pode fornecer uma visão sobre possíveis problemas que você possa ter com os dados.

   Algumas fontes de dados, como bancos de dados, não oferecem suporte à visualização de dados. Nesse caso, você pode executar um trabalho de perfil sem primeiro validar as regras de qualidade de dados. Em seguida, você pode obter informações sobre o esquema de dados e a distribuição de valores usando o perfil de dados.
6. Verifique a guia Recomendações, que lista algumas sugestões de regras que você pode usar ao criar seu conjunto de regras. Você pode selecionar todas, algumas ou nenhuma das recomendações.

   Depois de selecionar as recomendações relevantes, escolha Adicionar ao conjunto de regras.

   Isso adicionará regras ao seu conjunto de regras. Inspecione e modifique os parâmetros, se necessário. Observe que somente colunas de tipos simples, como string, números e booleanos, podem ser usadas nas regras de qualidade de dados.
7. Escolha Adicionar outra regra para adicionar uma regra não coberta pelas recomendações. Você pode alterar os nomes das regras para facilitar a interpretação posterior dos resultados da validação.

8. Use o escopo da verificação de qualidade de dados para escolher se colunas individuais serão selecionadas por cada verificação nessa regra ou se elas devem ser aplicadas a um grupo de colunas que você selecionar. Por exemplo, se seu conjunto de dados tiver várias colunas numéricas que devem ter valores entre 0 e 100, você pode definir a regra uma vez e selecionar todas essas colunas a serem verificadas por essa regra.
9. Se sua regra tiver mais de uma verificação, no menu suspenso Critérios de sucesso da regra, escolha se todas as verificações devem ser atendidas ou quais atendem aos critérios.
10. Selecione uma verificação que será executada para verificar essa regra no menu suspenso Verificação de qualidade de dados. Para obter mais informações sobre as verificações disponíveis, consulteCheques disponíveis.
11. Se você escolheu Verificação individual para cada coluna no escopo da verificação de qualidade de dados, escolha uma coluna. Selecione ou digite o nome da coluna para essa verificação.
12. Selecione os parâmetros de acordo com a verificação. Algumas condições aceitam somente valores personalizados fornecidos e algumas também oferecem suporte à referência a outra coluna.
13. Se você escolher verificar os valores da coluna, como a condição Contém, para valores de sequência de caracteres, poderá especificar o limite de “ultrapassagem”. Por exemplo, se você quiser que pelo menos 95% dos valores satisfaçam a condição, você precisa escolher Maior que igual como condição de limite, inserir 95 como limite e deixar “% (porcentagem) de linhas” na próxima lista suspensa na seção Limite. Ou se você não quiser mais do que 10 linhas em que o valor está ausente, a condição é verdadeira, selecione Menor que igual como condição, insira 10 em Limite e escolha linhas na próxima lista suspensa. Observe que você pode obter resultados diferentes se estiver usando amostras de tamanhos diferentes durante a validação.
14. Adicione mais regras, se necessário.
15. Escolha Criar conjunto de regras.

Criação de um trabalho de perfil usando um conjunto de regras

Depois de criar um conjunto de regras conforme descrito anteriormente, você será direcionado para a página Regras de qualidade de dados, que exibe todos os conjuntos de regras em sua conta.

Para criar um trabalho de perfil incluindo um conjunto de regras

1. Escolha o nome do conjunto de regras que você criou anteriormente para ver seus detalhes.
2. Escolha Criar perfil de trabalho com conjunto de regras.

O nome do Job é preenchido automaticamente, mas você pode alterá-lo conforme necessário.

3. Para Job run sample, você pode optar por executar todo o conjunto de dados ou um número limitado de linhas.

   Se você optar por executar um tamanho de amostra limitado, saiba que, para determinadas regras, os resultados podem ser diferentes em comparação com o conjunto de dados completo.

4. Para configurações de saída do trabalho, escolha um local do S3 para a saída do trabalho. Escolha qualquer pasta em um bucket nomeado do Amazon S3 ao qual você tenha acesso. Se você inserir um nome de pasta para esse bucket que não exista, essa pasta será criada.

   Após a conclusão bem-sucedida do trabalho de perfil, essa pasta conterá perfis do relatório de validação de dados e regras de qualidade de dados no formato JSON.

5. Em Regras de qualidade de dados, observe que seu conjunto de regras está listado em Nome do conjunto de regras de qualidade de dados.

6. Em Permissões, selecione ou crie uma função para conceder DataBrew acesso à leitura do local de entrada do Amazon S3 e à gravação no local de saída do trabalho. Se você não tiver uma função pronta, selecione Criar nova função do IAM.

7. Modifique quaisquer outras configurações opcionais conforme descrito emCriando e trabalhando com AWS Glue DataBrew empregos de perfil, se necessário.

8. Escolha Criar e executar tarefa.

Inspecionando os resultados da validação e atualizando as regras de qualidade de dados

Depois que seu trabalho de perfil for concluído, você poderá visualizar os resultados da validação de suas regras de qualidade de dados e, conforme necessário, atualizá-las.

Para visualizar os dados de validação de suas regras de qualidade de dados

1. No DataBrew console, escolha Exibir perfil de dados. Isso exibe a guia Visão geral do perfil de dados do seu conjunto de dados.

2. Escolha a guia Regras de qualidade de dados. Nessa guia, você pode ver os resultados de todas as suas regras de qualidade de dados.

3. Selecione uma regra individual para obter mais detalhes sobre essa regra.

Para qualquer regra que falhou na validação, você pode fazer as correções necessárias.

Para atualizar suas regras de qualidade de dados

1. No painel de navegação, escolha DQ RULES.
2. Em Nome do conjunto de regras de qualidade de dados, escolha o conjunto de dados que contém as regras que você planeja editar.
3. Escolha a regra que você deseja alterar e, em seguida, escolha Editar.
4. Faça as correções necessárias e escolha Atualizar conjunto de regras.
5. Execute novamente o trabalho. Repita esse processo até que todas as validações sejam aprovadas.

Cheques disponíveis

A tabela a seguir lista referências para todas as condições disponíveis que podem ser usadas em suas regras. Observe que condições agregadas não podem ser combinadas com condições não agregadas na mesma regra.

> **Note**
>
> Para usuários do SDK, para aplicar a mesma regra a várias colunas, use o ColumnSelectorsatributo de uma regra e especifique as colunas validadas usando seus nomes ou uma expressão regular. Nesse caso, você deve usar implícito. CheckExpression Por exemplo, `“> :val”` para comparar valores em cada uma das colunas selecionadas com o valor fornecido. DataBrew usa sintaxe implícita para definir FilterExpressionem conjuntos de dados dinâmicos. Se você quiser especificar colunas para cada verificação individualmente, não defina o ColumnSelectorsatributo. Em vez disso, forneça uma expressão explícita. Por exemplo, `“:col > :val”` como CheckExpressionem uma regra.

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK
Condições agregadas do	Número de linhas		Comparação numérica com o	`"CheckExpression":`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK
			valor personalizado	`"AGG(ROWS_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
conjunto de dados	Número de colunas		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Linhas duplicadas		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

Condições de estatísticas de colunas agregadas	Valores ausentes		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores duplicados		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores válidos		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
				
				or
				
				`"CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores distintos		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(DISTINCT_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "1000"}`
				
				or
				
				`"CheckExpression": "AGG(DISTINCT_VALUES_PERCENTAGE) >= :val", "SubstitutionMap": {":val", "50"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores exclusivos		Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores discrepantes	Z-score limiar	Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Estatísticas de distribuição de valor	Nome da estatística (veja a tabela a seguir)	Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Veja a tabela a seguir para `STAT_NAME` valores possíveis

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Estatística numérica	Nome da estatística (veja a tabela a seguir)	Comparação numérica com o valor personalizado	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Veja a tabela a seguir para `STAT_NAME` valores possíveis

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK
Não agregado (aceita limite)	O valor é exatamente		Comparação exata com uma lista de valores	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	O valor não é exatamente		O valor não deve corresponder exatamente a nenhum valor de uma lista	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores de string		Comparação de string com valor personalizado ou outra coluna de string	`"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"}`
				
				or
				
				`"CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Valores numéricos		Comparação numérica com valor personalizado ou outra coluna numérica	`"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"}`
				
				or
				
				`"CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}`

Tipo de condição	Verificação de qualidade de dados	Parâmetros adicionais	Tipo de comparação	Exemplo de sintaxe do SDK

	Comprimento da cadeia de valores		Comparação numérica com valor personalizado ou outra coluna numérica	`"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"}`
				
				or
				
				`"CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}`

Comparações numéricas

DataBrew suporta as seguintes operações para comparação numérica: Is equals (= =), Is not equals (! =), Menor que (<), Menor que é igual (< =), Maior que (>), Maior que é igual a (> =) e Está entre (is_between:val1 e:val2).

Comparações de strings

As seguintes comparações de strings são suportadas: Começa com, Não começa com, Termina com, Não termina com, Contém, Não contém, É igual, Não é igual, Corresponde, Não corresponde.

A tabela a seguir exibe as estatísticas disponíveis que você pode usar para estatísticas de distribuição de valor e estatísticas numéricas:

Verificação de qualidade de dados	Nome da estatística	Parâmetros adicionais	Sintaxe do SDK
Estatísticas de distribuição de valor	Mín.		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	Máx		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	Mediana		`"CheckExpression": "AGG(MEDIAN) >= :val", "Substitu`

Verificação de qualidade de dados	Nome da estatística	Parâmetros adicionais	Sintaxe do SDK
			`tionMap": {":val", "50"}`
	Média		`"CheckExpression": "AGG(MEAN) <= :val", "SubstitutionMap": {":val", "10"}`
	Modo		`"CheckExpression": "AGG(MODE) > :val", "SubstitutionMap": {":val", "0"}`
	Desvio padrão		`"CheckExpression": "AGG(STANDARD_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`

Verificação de qualidade de dados	Nome da estatística	Parâmetros adicionais	Sintaxe do SDK
	Entropia		`"CheckExpression": "AGG(ENTROPY) > :val", "SubstitutionMap": {":val", "0"}`
Estatística numérica	Soma		`"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}`
	Curtose		`"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}`
	Distorção		`"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}`

Verificação de qualidade de dados	Nome da estatística	Parâmetros adicionais	Sintaxe do SDK
	Variação		`"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}`
	Desvio absoluto		`"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Quantil	Quantil: um de '0,25', '0,5', '0,75'	`"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}`

Criando e usando AWS Glue DataBrew projetos

Em AWS Glue DataBrew, um projeto é a peça central de seus esforços de análise e transformação de dados.

Ao criar um projeto, você reúne dois componentes fundamentais:

- Um conjunto de dados, para fornecer acesso somente de leitura aos seus dados de origem. Para obter mais informações, consulte Conectando-se aos dados com AWS Glue DataBrew.
- Uma receita para aplicar transformações DataBrew de dados ao conjunto de dados. Para obter mais informações, consulte Criando e usando AWS Glue DataBrew recipes.

O DataBrew console apresenta seu projeto em uma interface de usuário altamente interativa e intuitiva. Ele incentiva você a experimentar centenas de transformações de dados, para que você possa aprender como elas funcionam e o efeito que elas têm nos seus dados.

Os dados que você vê na visualização do projeto são uma amostra do seu conjunto de dados. Como os conjuntos de dados podem ser muito grandes, com milhares ou até milhões de linhas, o uso de uma amostra ajuda a garantir que o DataBrew console permaneça responsivo enquanto você transforma os dados de amostra de várias maneiras. Por padrão, a amostra consiste nas primeiras 500 linhas de dados do conjunto de dados. Você pode escolher configurações diferentes para o tamanho da amostra e quais linhas serão escolhidas.

À medida que você transforma os dados de amostra, DataBrew ajuda a criar e refinar a receita do projeto — uma série passo a passo das transformações que você aplicou até agora. Sua receita de trabalho em andamento é salva automaticamente, para que você possa sair da visualização do projeto a qualquer momento, retornar mais tarde e continuar de onde parou.

Quando sua receita estiver pronta para uso, você poderá publicá-la. A publicação de uma receita a disponibiliza para o subsistema de DataBrew tarefas, onde você pode aplicar a receita a todo o conjunto de dados ou criar um perfil de dados abrangente que permite entender a estrutura, o conteúdo e as características estatísticas dos seus dados.

Tópicos

Criação de um projeto

Use o procedimento a seguir para criar um projeto.

Para criar um projeto

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console.
2. No painel de navegação, escolha PROJETOS. Em seguida, escolha Criar projeto.
3. Insira um nome para o projeto. Em seguida, escolha uma receita para anexar ao seu projeto:

   - Escolha Criar nova receita se você estiver começando do início. Isso cria uma receita nova e vazia e a anexa ao seu projeto.
   - Escolha Editar receita existente se você tiver uma receita publicada anteriormente que deseja usar para este projeto. Se a receita estiver atualmente anexada a outro projeto ou tiver algum trabalho definido para ela, você não poderá usá-la em seu novo projeto. Escolha Procurar receitas para ver quais receitas estão disponíveis.
   - Escolha Importar etapas da receita se você tiver uma receita existente publicada anteriormente e quiser importar suas etapas e, em seguida, faça o seguinte:
     1. Escolha Procurar receitas para ver quais receitas estão disponíveis.
     2. Escolha a versão publicada da receita que você deseja usar. Uma receita pode ter várias versões, dependendo da frequência com que você a publicou enquanto trabalhava na visualização do projeto.
     3. Escolha Exibir etapas da receita para examinar as transformações de dados na receita.
4. Depois de criar uma receita, escolha o conjunto de dados com o qual você deseja trabalhar no painel Selecionar um conjunto de dados:

   - Meus conjuntos de dados — Escolha um conjunto de dados que você criou anteriormente. Para mais informações, consulte Criação de um projeto.)
   - Arquivos de amostra — Crie um novo conjunto de dados com base nos dados de amostra mantidos pelo AWS. Esses dados de amostra são uma ótima maneira de explorar o que DataBrew você pode fazer, sem precisar fornecer seus próprios dados. Certifique-se de inserir um nome para seu conjunto de dados.
   - Novo conjunto de dados — Crie um novo conjunto de dados. Para obter mais informações, consulte Criação de um projeto.
5. Para permissões de acesso, escolha uma função AWS Identity and Access Management(IAM) que permita ler DataBrew a partir do seu local de entrada do Amazon S3. Para um local

do S3 pertencente à sua AWS conta, você pode escolher a função gerenciada pelo `AwsGlueDataBrewDataAccessRole` serviço. Isso permite DataBrew acessar os recursos do S3 que você possui.

6. No painel Amostragem, você pode encontrar opções DataBrew para criar uma amostra de dados do seu conjunto de dados.

   Em Tipo, escolha como obter DataBrew as linhas do seu conjunto de dados:

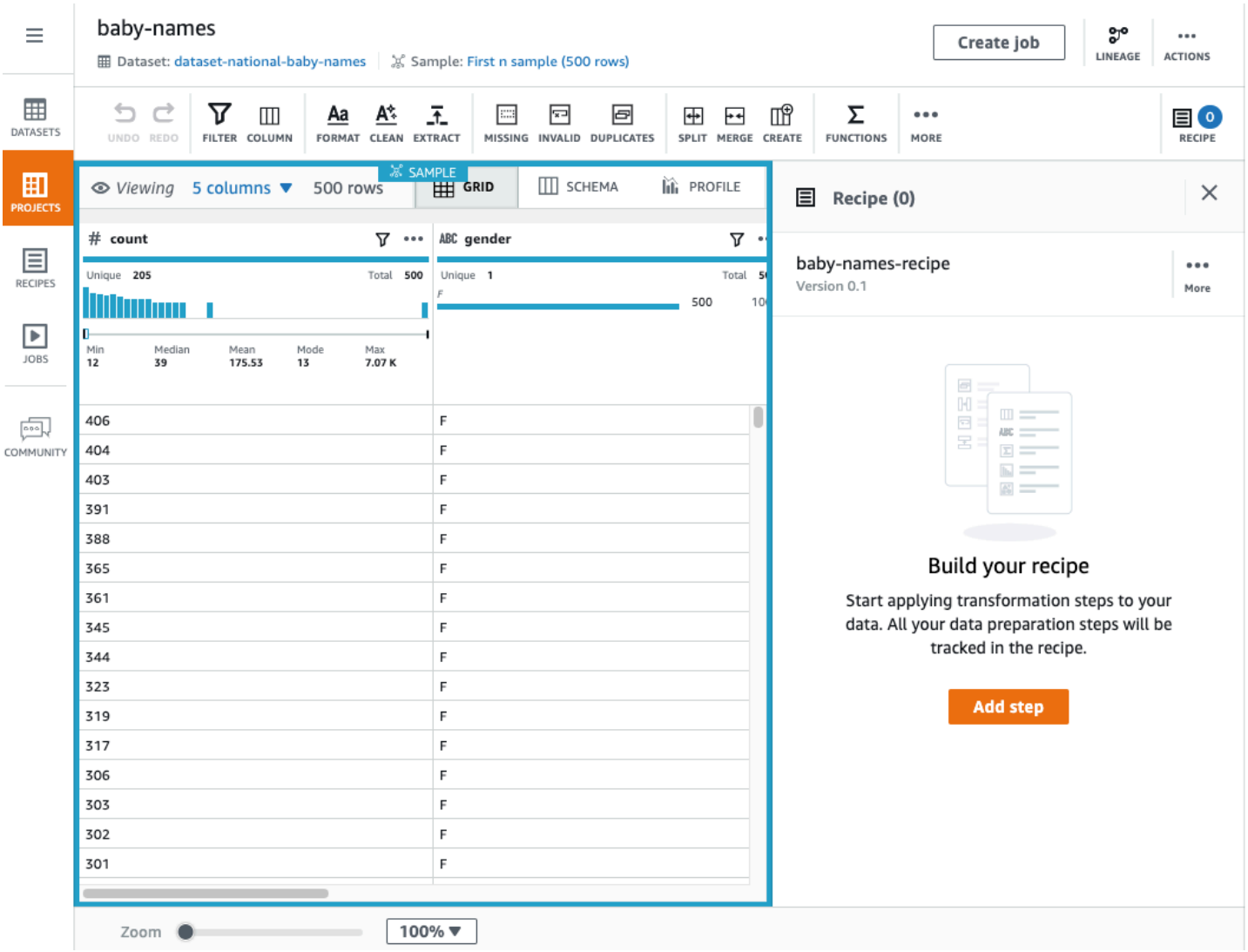
   - Use Primeiras n linhas para criar uma amostra com base nas primeiras linhas do conjunto de dados.
   - Use Linhas aleatórias para criar uma amostra com base em uma seleção aleatória de linhas no conjunto de dados.
   - Escolha o número de linhas a serem exibidas na amostra: 500, 1.000, 2.500 ou um tamanho de amostra personalizado, até um máximo de 5.000 linhas. Um tamanho de amostra menor permite DataBrew realizar transformações mais rapidamente, economizando tempo ao desenvolver sua receita. Um tamanho amostral maior reflete com mais precisão a composição dos dados de origem subjacentes. No entanto, a inicialização da sessão do projeto e as transformações interativas são mais lentas.
7. (Opcional) Escolha Tags para anexar tags ao seu conjunto de dados.

   As tags são rótulos simples que consistem em uma chave definida pelo usuário e um valor opcional que pode facilitar o gerenciamento, a pesquisa e a filtragem de DataBrew projetos por finalidade, proprietário, ambiente ou outros critérios.
8. Quando as configurações estiverem como você deseja, escolha Criar trabalho.

DataBrew cria um novo conjunto de dados, se necessário, cria uma nova receita, se necessário, cria a amostra de dados e cria uma sessão de projeto interativa. Esse processo pode levar alguns minutos para ser concluído. Quando o projeto estiver pronto para uso, você poderá começar a trabalhar com a amostra de dados.

Visão geral de uma sessão de DataBrew projeto

Em uma sessão de DataBrew projeto, você trabalha em um espaço de trabalho interativo.



O painel esquerdo mostra a visualização atual dos seus dados. O painel direito mostra a receita de transformação do projeto, que está vazia no momento.

No canto superior direito da grade de dados, há três guias:`GRID`, e. `SCHEMA` `PROFILE` A escolha de uma dessas guias exibe uma visualização correspondente na área de trabalho; essas visualizações são descritas a seguir.

Visualização em grade

A visualização em grade é a visualização padrão, na qual a amostra é mostrada em formato tabular. Use o procedimento a seguir para uma breve explicação da visualização em grade.

Para fazer uma explicação passo a passo da visualização em grade

1. Comece visualizando todo o espaço:

a. Role para a esquerda e para a direita para ver todas as colunas.

b. Role para cima e para baixo para ver todos os valores dos dados.

c. Use o controle de zoom na parte inferior da área de trabalho para ajustar o nível de ampliação da grade.

2. No canto superior direito, veja quantas colunas da amostra são mostradas e o número atual de linhas na amostra.

   Para alterar quais colunas são mostradas, escolha o link N colunas (onde N é o número de colunas exibidas atualmente). Escolha as colunas que você deseja e escolha Mostrar colunas selecionadas.

3. Agora você pode começar a experimentar com DataBrew transformações. Faça o seguinte:

   a. Na barra de ferramentas de transformação, escolha Escolher formato, Alterar para maiúsculas.

   b. Em Coluna de origem, escolha uma coluna que contenha dados de caracteres.

   c. Deixe as outras configurações nos valores padrão.

   d. Para ver a aparência dos dados transformados, escolha Visualizar alterações. Em seguida, para adicionar essa transformação à sua receita, escolha Aplicar.

   Sempre que você aplica uma transformação de dados, DataBrew adicione-a à cópia de trabalho da sua receita. Isso aparece no lado direito do seu espaço de trabalho.

4. Faça o seguinte:

   a. Na barra de ferramentas de transformação, escolha Criar, com base em uma função.

   b. Em Selecionar uma função, escolha`SQUARE ROOT`.

   c. Em Coluna de origem, escolha uma coluna que contenha dados numéricos.

   d. Deixe as outras configurações em seus padrões,.

   e. Escolha Visualizar alterações para ver a aparência dos dados transformados. Em seguida, para adicionar essa transformação à sua receita, escolha Aplicar.

5. Feche o painel de receitas no canto superior direito escolhendo RECEITA. Para expandir o painel de receitas, escolha RECEITA novamente.

Publicando uma nova versão da sua receita

À medida que você continua aplicando as transformações, o número de etapas na receita aumenta. A qualquer momento, você pode publicar uma nova versão da sua receita. A publicação de uma receita a torna disponível em outros lugares DataBrew. Ao fazer isso, você pode executar um trabalho de receita para transformar todo o conjunto de dados, em vez de transformar somente a amostra de dados do projeto.

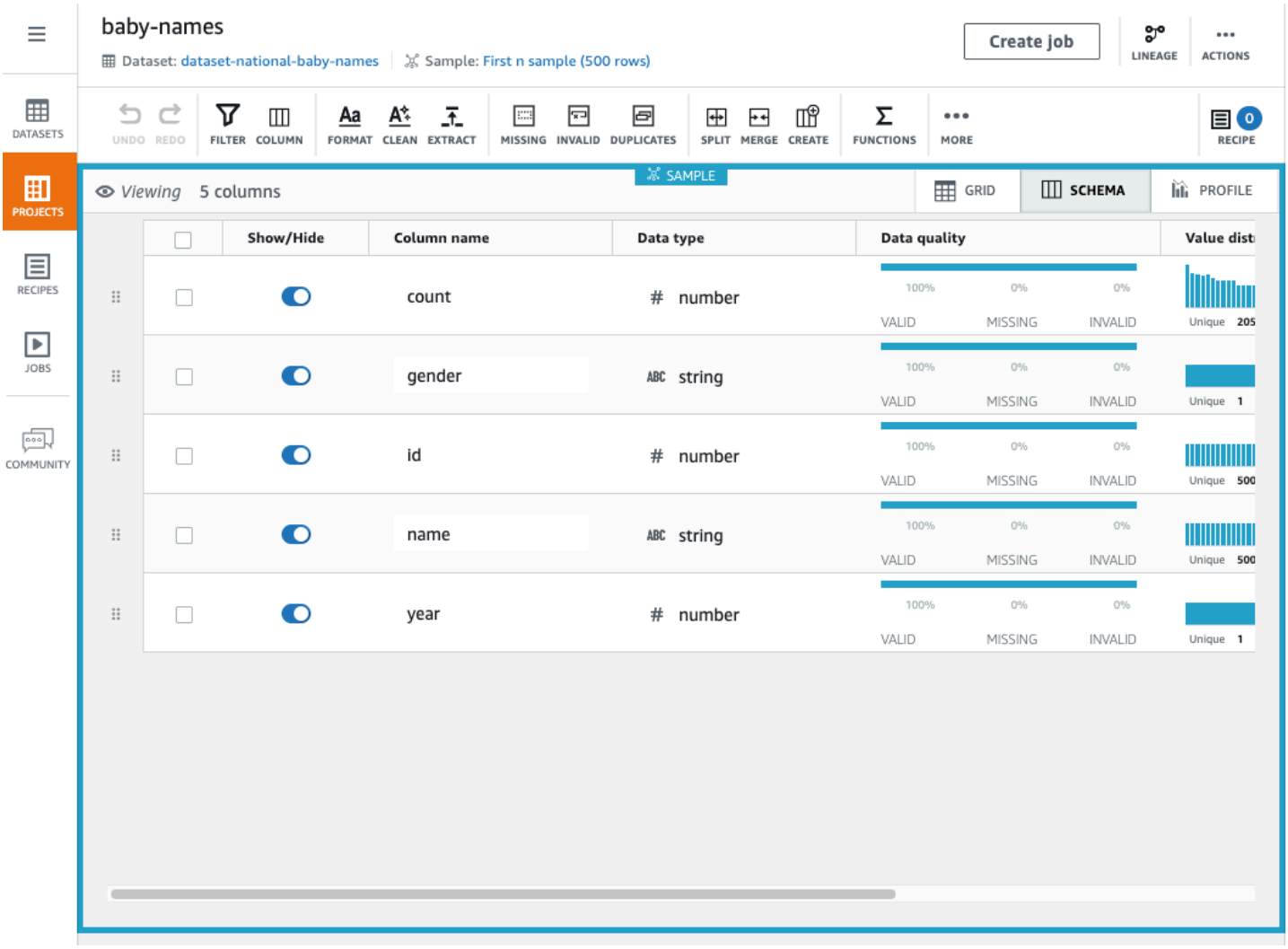
A publicação de receitas também incentiva uma abordagem incremental e iterativa para o desenvolvimento de receitas: você pode publicar novas versões da receita à medida que avança, para poder voltar à “última versão válida” da receita, se necessário.

Para publicar uma nova versão de uma receita

- No painel de receitas, escolha Publicar. Insira uma descrição para essa versão da receita e escolha Publicar.

Visualização do esquema

Se você escolher a guia ESQUEMA, a exibição será alterada, conforme mostrado na captura de tela a seguir.



Na visualização do esquema, você pode ver estatísticas sobre os valores dos dados em cada coluna.

Na coluna da extrema esquerda, ao lado de Show/Hide, escolha qualquer uma das colunas de dados. O painel de detalhes da coluna aparece à direita. Esse painel mostra um resumo das estatísticas dos valores da coluna.

Você pode renomear uma coluna inserindo um novo nome para Nome da coluna.

Você pode reorganizar a ordem das colunas arrastando e soltando as colunas.

Visualização do perfil

Se você escolher a guia PERFIL, poderá ver informações volumétricas detalhadas sobre seu projeto. Antes de fazer isso, você executa um DataBrew trabalho para criar o perfil.

Para dar uma olhada na visualização do perfil

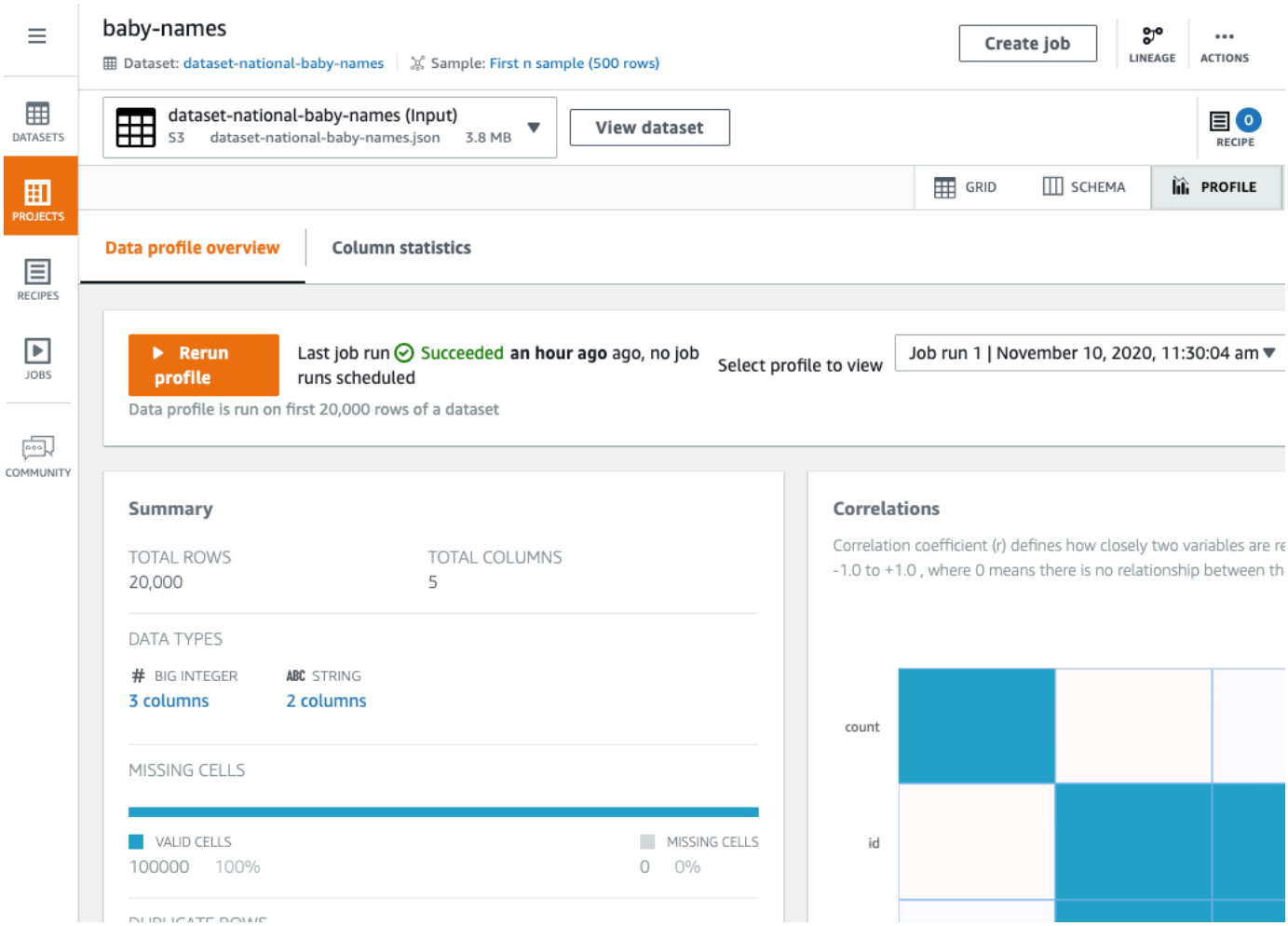
1. Escolha Criar trabalho e insira um nome para seu trabalho.
2. Em Saída de Job, escolha CSV para o tipo de arquivo.
3. Encontre ou crie um bucket e uma pasta do Amazon S3 em sua AWS conta onde você deseja que a saída do DataBrew trabalho seja gravada:

   - Se você já tiver esse bucket e pasta do Amazon S3, escolha Procurar e localize-os. Verifique se você tem permissões de gravação para ambos.
   - Se você não tiver esse bucket e essa pasta do Amazon S3, crie-os:

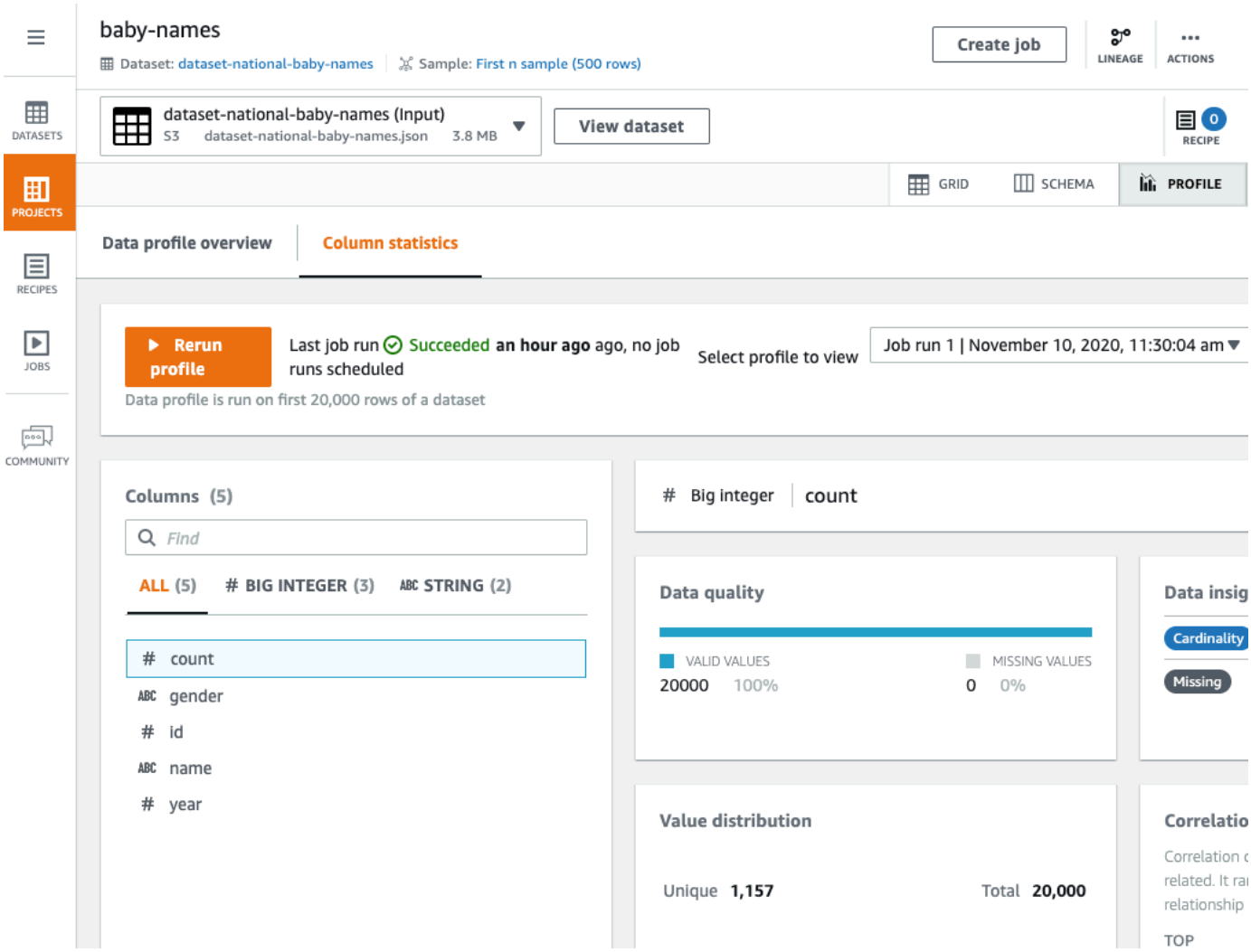
     1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
     2. Se você não tiver um bucket do Amazon S3, escolha Create bucket. Em Nome do bucket, insira um nome exclusivo para seu novo bucket. Selecione Criar bucket.
     3. Na lista de compartimentos, escolha aquele que você deseja usar.
     4. Selecione Criar pasta. Em Nome da pasta`databrew-output`, insira e escolha Criar pasta.
4. Para permissões de acesso, escolha uma função do IAM que permita DataBrew gravar no seu local de saída do Amazon S3.

   Para um local do S3 pertencente à sua AWS conta, você pode escolher a função gerenciada pelo `AwsGlueDataBrewDataAccessRole` serviço. Isso permite DataBrew acessar os recursos do S3 que você possui.
5. Deixe as outras configurações em seus padrões e escolha Criar e executar tarefa.
6. Depois que o trabalho for concluído, o espaço de trabalho exibirá um resumo gráfico do perfil de dados.

   A guia Visão geral do perfil de dados mostra um resumo de alto nível das características dos seus dados, conforme mostrado na captura de tela a seguir.



A guia Estatísticas da coluna mostra um detalhamento, coluna por coluna, dos valores dos dados:



Excluir um projeto

Se você não precisar mais de um projeto, poderá excluí-lo.

Para excluir um projeto

1. No painel de navegação, escolha PROJETOS.
2. Escolha o projeto que você deseja excluir e, em Ações, escolha Excluir. .

Criando e usando AWS Glue DataBrew recipes

Em DataBrew, uma receita é um conjunto de etapas de transformação de dados. Você pode aplicar essas etapas a uma amostra dos seus dados ou aplicar a mesma receita a um conjunto de dados.

A maneira mais fácil de desenvolver uma receita é criar um DataBrew projeto, no qual você pode trabalhar interativamente com uma amostra dos seus dados. Para obter mais informações, consulte. Criando e usando AWS Glue DataBrew projetos Como parte do fluxo de trabalho de criação do projeto, uma nova receita (vazia) é criada e anexada ao projeto. Em seguida, você pode começar a criar sua receita adicionando transformações de dados.

> **Note**
>
> Você pode incluir até 100 transformações de dados em uma única DataBrew receita.

Ao continuar desenvolvendo sua receita, você pode salvar seu trabalho publicando a receita. DataBrew mantém uma lista das versões publicadas da sua receita. Você pode usar qualquer versão publicada em um trabalho de receita para executar a receita (em um trabalho de receita) e transformar seu conjunto de dados. Você também pode baixar uma cópia das etapas da receita para poder reutilizá-la em outros projetos ou em outras transformações do conjunto de dados.

Você também pode desenvolver DataBrew receitas programaticamente, usando o AWS Command Line Interface(AWS CLI) ou um dos SDKs.AWS Na DataBrew API, as transformações são conhecidas como ações de receita.

> **Note**
>
> Em uma sessão de DataBrew projeto interativa, cada transformação de dados aplicada resulta em uma chamada para a DataBrew API. Essas chamadas de API ocorrem automaticamente, sem que você precise conhecer os detalhes dos bastidores.

Mesmo que você não seja programador, é útil entender a estrutura de uma receita e como DataBrew organiza as ações da receita.

Tópicos

- Publicando uma nova versão da receita

- Definindo uma estrutura de receita

Publicando uma nova versão da receita

Você publica novas versões de uma receita em uma sessão interativa DataBrew do projeto.

Para publicar uma nova versão da receita

1. No painel de receitas, escolha Publicar.
2. Insira uma descrição para essa versão da receita e escolha Publicar.

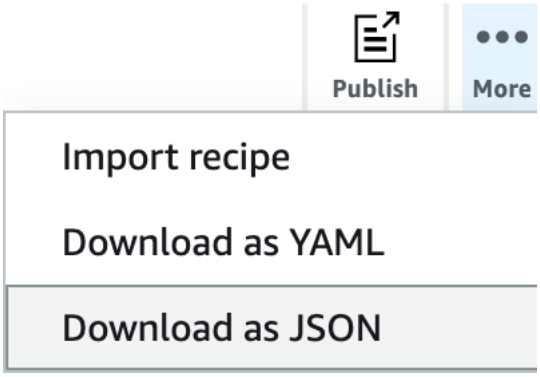
Você pode ver todas as suas receitas publicadas e suas versões escolhendo PROJETOS no painel de navegação.

Definindo uma estrutura de receita

Ao criar um projeto pela primeira vez usando o DataBrew console, você define uma receita a ser associada a esse projeto. Se você não tiver uma receita existente, o console cria uma para você.

Ao trabalhar com seu projeto no console, você usa a barra de ferramentas de transformação para aplicar ações aos dados de amostra do seu conjunto de dados. O console mostra as etapas da receita e a ordem dessas etapas à medida que você continua criando a receita. Você pode iterar e refinar a receita até ficar satisfeito com as etapas.

EmConceitos básicos de AWS Glue DataBrew, você cria uma receita para transformar um conjunto de dados de jogos de xadrez famosos. Você pode baixar uma cópia das etapas da receita escolhendo Baixar como JSON ou Baixar como YAML, conforme mostrado na captura de tela a seguir.



O arquivo JSON baixado contém ações de receita correspondentes às transformações que você adicionou à sua receita.

Uma nova receita não tem etapas. Você pode representar uma nova receita como uma lista JSON vazia, conforme mostrado a seguir.

```
[ ]
```

A seguir está um exemplo desse arquivo, para`chess-project-recipe`. A lista JSON contém vários objetos que descrevem as etapas da receita. Cada objeto na lista JSON está entre colchetes (). { } As linhas JSON são delimitadas por vírgulas.

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

É mais fácil ver que cada ação é uma linha individual se adicionarmos apenas novas linhas para novas ações, conforme mostrado a seguir.

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

As ações são executadas sequencialmente, na mesma ordem do arquivo:

- `REMOVE_VALUES`— Para filtrar todos os jogos em que a classificação de um jogador é inferior a 1.800, a classificação mínima é necessária para ser um jogador de xadrez Classe A. Há duas ocorrências dessa ação: uma para remover jogadores do lado preto que não sejam pelo menos jogadores da Classe A e outra para remover jogadores do lado branco que não estejam nesse nível.
- `GROUP_BY`— Para resumir os dados. Nesse caso, GROUP_BY classifica as linhas em grupos com base nos valores de `winner` (`blacke`). `white` Cada um desses grupos é então dividido ainda mais, classificando as linhas em subgrupos com base nos valores de `victory_status` (`mate`, `resignoutoftime`, `edraw`). Finalmente, o número de ocorrências para cada subgrupo é contado. O resumo resultante então substitui a amostra de dados original.
- `REMOVE_VALUES`— Para excluir os resultados dos jogos que terminaram com`draw`.
- `REPLACE_TEXT`— Para modificar os valores de`victory_status`. Há três ocorrências dessa ação — uma para cada`mate`, e. `resign` `oufoftime`

Em uma sessão de DataBrew projeto interativa, cada uma `RecipeAction` corresponde a uma transformação de dados que você aplica a uma amostra de dados.

DataBrew fornece mais de 200 ações de receitas. Para obter mais informações, consulte Etapa da receita e referência da função.

Usar condições

Você pode usar condições para restringir o escopo de uma ação de receita. As condições são usadas em transformações que filtram os dados, por exemplo, removendo linhas indesejadas com base em um valor de coluna específico.

Vamos dar uma olhada mais de perto nas ações de uma receita`chess-project-recipe`.

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

```
        "Value": "1800",
        "TargetColumn": "black_rating"
      }
    ]
  }
```

Essa transformação lê os valores na `black_rating` coluna. A `ConditionExpressions` lista determina os critérios de filtragem: qualquer linha com um `black_rating` valor inferior a 1.800 é removida do conjunto de dados.

Uma transformação subsequente na receita faz a mesma coisa, pois`white_rating`. Dessa forma, os dados são limitados aos jogos em que cada jogador (preto ou branco) é classificado na Classe A ou superior.

Aqui está outro exemplo de uma condição, aplicada a uma coluna de dados de caracteres.

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

Essa transformação lê os valores na `winner` coluna, procurando o valor `draw` e removendo essas linhas. Dessa forma, os dados são limitados apenas aos jogos em que houve um vencedor claro.

DataBrew suporta as seguintes condições:

- `IS`— O valor na coluna é igual ao valor fornecido na condição.
- `IS_NOT`— O valor na coluna não é o mesmo que o valor fornecido na condição.
- `IS_BETWEEN`— O valor na coluna está entre os `LESS_THAN_EQUAL` parâmetros `GREATER_THAN_EQUAL` e.

- `CONTAINS`— O valor da string na coluna contém o valor que foi fornecido na condição.
- `NOT_CONTAINS`— O valor na coluna não contém a cadeia de caracteres fornecida na condição.
- `STARTS_WITH`— O valor na coluna começa com a cadeia de caracteres fornecida na condição.
- `NOT_STARTS_WITH`— O valor na coluna não começa com a cadeia de caracteres fornecida na condição.
- `ENDS_WITH`— O valor na coluna termina com a cadeia de caracteres fornecida na condição.
- `NOT_ENDS_WITH`— O valor na coluna não termina com a cadeia de caracteres fornecida na condição.
- `LESS_THAN`— O valor na coluna é menor que o valor fornecido na condição.
- `LESS_THAN_EQUAL`— O valor na coluna é menor ou igual ao valor fornecido na condição.
- `GREATER_THAN`— O valor na coluna é maior do que o valor fornecido na condição.
- `GREATER_THAN_EQUAL`— O valor na coluna é maior ou igual ao valor fornecido na condição.
- `IS_INVALID`— O valor na coluna tem um tipo de dados incorreto.
- `IS_MISSING`— Não há valor na coluna.

Criação, execução e agendamento AWS Glue DataBrew jobs

AWS Glue DataBrew tem um subsistema de tarefas que serve a duas finalidades:

1. Aplicar uma receita de transformação de dados a um DataBrew conjunto de dados. Você faz isso com um trabalho de DataBrew receita.
2. Analisar um conjunto de dados para criar um perfil abrangente dos dados. Você faz isso com um trabalho DataBrew de perfil.

Tópicos

Criando e trabalhando com AWS Glue DataBrew trabalhos de receita

Use um trabalho de DataBrew receita para limpar e normalizar os dados em um DataBrew conjunto de dados e gravar o resultado em um local de saída de sua escolha. A execução de um trabalho de receita não afeta o conjunto de dados nem os dados de origem subjacentes. Quando uma tarefa é executada, ela se conecta aos dados de origem somente para leitura. A saída do trabalho é gravada em um local de saída que você define no Amazon S3, no ou em um banco de AWS Glue Data Catalog dados JDBC compatível.

Use o procedimento a seguir para criar um trabalho de DataBrew receita.

Para criar um trabalho de receita

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console em https://console.aws.amazon.com/databrew/.
2. Escolha TRABALHOS no painel de navegação, escolha a guia Trabalhos de receita e, em seguida, escolha Criar trabalho.
3. Insira um nome para seu trabalho e, em seguida, escolha Criar um trabalho de receita.

4. Em Job input, insira detalhes sobre o trabalho que você deseja criar: o nome do conjunto de dados a ser processado e a receita a ser usada.

   Um trabalho de receita usa uma DataBrew receita para transformar um conjunto de dados. Para usar uma receita, certifique-se de publicá-la primeiro.

5. Defina as configurações de saída do trabalho.

   Forneça um destino para a produção do seu trabalho. Se você não tiver uma DataBrew conexão configurada para seu destino de saída, configure-a primeiro na guia DATASETS, conforme descrito emConexões suportadas para fontes e saídas de dados. Escolha um dos seguintes destinos de saída:

   - Amazon S3, com ou sem suporte AWS Glue Data Catalog
   - Amazon Redshift, com ou sem suporte AWS Glue Data Catalog
   - JDBC
   - Mesas Snowflake
   - Tabelas de banco de dados do Amazon RDS com AWS Glue Data Catalog suporte. As tabelas de banco de dados do Amazon RDS oferecem suporte aos seguintes mecanismos de banco de dados:
     - Amazon Aurora
     - MySQL
     - Oracle
     - PostgreSQL
     - Microsoft SQL Server
   - Amazon S3 com AWS Glue Data Catalog suporte.

   Para AWS Glue Data Catalog saída baseada em AWS Lake Formation, DataBrew suporta somente a substituição de arquivos existentes. Nessa abordagem, os arquivos são substituídos para manter intactas as permissões existentes do Lake Formation para sua função de acesso a dados. Além disso, DataBrew dá precedência à localização do Amazon S3 na tabela AWS Glue Data Catalog. Portanto, você não pode substituir a localização do Amazon S3 ao criar um trabalho de receita.

   Em alguns casos, a localização do Amazon S3 na saída do trabalho difere da localização do Amazon S3 na tabela do catálogo de dados. Nesses casos, DataBrew atualiza a definição do

trabalho automaticamente com a localização do Amazon S3 na tabela do catálogo. Ele faz isso quando você atualiza ou inicia seus trabalhos existentes.

6. Somente para destinos de saída do Amazon S3, você tem mais opções:

   a. Escolha um dos formatos de saída de dados disponíveis para o Amazon S3, compactação opcional e um delimitador personalizado opcional. Os delimitadores compatíveis para arquivos de saída são os mesmos de entrada: vírgula, dois pontos, ponto e vírgula, barra vertical, tabulação, circunflexo, barra invertida e espaço. Para obter detalhes sobre a formatação, consulte a tabela a seguir.

Formato	Extensão de arquivo (não compactada)	Extensões de arquivo (compactadas)
Comma-separated valores	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated valores	`.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	Não compatível	`.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`

Formato	Extensão de arquivo (não compactada)	Extensões de arquivo (compactadas)
XML	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`
JSON (somente no formato de linhas JSON)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	Não compatível	Não aplicável

b. Escolha se deseja gerar um único arquivo ou vários arquivos. Há três opções para saída de arquivos com o Amazon S3:

- Geração automática de arquivos (recomendado) — DataBrew Determina o número ideal de arquivos de saída.
- Saída de arquivo único — Faz com que um único arquivo de saída seja gerado. Essa opção pode resultar em tempo adicional de execução do trabalho porque o pós-processamento é necessário.
- Saída de vários arquivos — Você especifica o número de arquivos para a saída do seu trabalho. Os valores válidos são de 2 a 999. Menos arquivos do que você especifica podem ser gerados se o particionamento de colunas for usado ou se o número de linhas na saída for menor que o número de arquivos que você especificar.

c. (Opcional) Escolha o particionamento de colunas para a saída do trabalho de receita.

O particionamento de colunas fornece outra maneira de particionar a saída do seu trabalho de receita em vários arquivos. O particionamento de colunas pode ser usado com uma saída nova ou existente do Amazon S3 ou com a nova saída do Catálogo de Dados do Amazon S3. Ele não pode ser usado com as tabelas existentes do Catálogo de Dados Amazon S3. Os arquivos de saída são baseados nos valores dos nomes das colunas que você especifica. Se os nomes das colunas que você especificar forem exclusivos, os

caminhos de pasta do Amazon S3 resultantes serão baseados na ordem dos nomes das colunas.

Para obter um exemplo de particionamento de colunasExemplo de particionamento de colunas, consulte a seguir.

7. (Opcional) Escolha Ativar criptografia para saída do trabalho para criptografar a saída do trabalho que DataBrew grava no seu local de saída e, em seguida, escolha o método de criptografia:

   - Use SSE-S3 criptografia — A saída é criptografada usando criptografia do lado do servidor com chaves de criptografia gerenciadas pelo Amazon S3.
   - Use AWS Key Management Service(AWS KMS) — A saída é criptografada usando AWS KMS. Para usar essa opção, escolha o Amazon Resource Name (ARN) da AWS KMS chave que você deseja usar. Se você não tiver uma AWS KMS chave, poderá criar uma escolhendo Criar uma AWS KMS chave.

8. Para permissões de acesso, escolha uma função AWS Identity and Access Management(IAM) que permita DataBrew gravar em seu local de saída. Para um local pertencente à sua AWS conta, você pode escolher a função `AwsGlueDataBrewDataAccessRole` gerenciada pelo serviço. Isso permite DataBrew acessar AWS os recursos que você possui.

9. No painel Configurações avançadas do trabalho, você pode escolher mais opções de como seu trabalho deve ser executado:

   - Número máximo de unidades — DataBrew processa trabalhos usando vários nós de computação, executados em paralelo. O número padrão de nós é 5. O número máximo de nós é 149.
   - Tempo limite do trabalho — Se um trabalho levar mais do que o número de minutos que você definiu aqui para ser executado, ele falhará com um erro de tempo limite. O valor padrão é 2.880 minutos ou 48 horas.
   - Número de novas tentativas — Se um trabalho falhar durante a execução, DataBrew pode tentar executá-lo novamente. Por padrão, o trabalho não é repetido.
   - Habilitar Amazon CloudWatch Logs para trabalho — Permite DataBrew publicar informações de diagnóstico no CloudWatch Logs. Esses registros podem ser úteis para solucionar problemas ou para obter mais detalhes sobre como o trabalho é processado.

10. Para trabalhos agendados, você pode aplicar um cronograma de DataBrew trabalho para que seu trabalho seja executado em um horário específico ou de forma recorrente. Para obter mais informações, consulte Automatizando a execução de trabalhos com um cronograma.

11. Quando as configurações estiverem como você deseja, escolha Criar trabalho. Ou, se você quiser executar o trabalho imediatamente, escolha Criar e executar o trabalho.

Você pode monitorar o progresso do seu trabalho verificando seu status enquanto o trabalho está em execução. Quando a execução do trabalho é concluída, o status muda para Bem-sucedido. A saída do trabalho agora está disponível no local de saída escolhido.

DataBrew salva sua definição de tarefa, para que você possa executar a mesma tarefa posteriormente. Para executar novamente um trabalho, escolha Trabalhos no painel de navegação. Escolha o trabalho com o qual você deseja trabalhar e, em seguida, escolha Executar trabalho.

Exemplo de particionamento de colunas

Como exemplo de particionamento de colunas, suponha que você especifique três colunas, cada linha contendo um dos dois valores possíveis. A `Dept` coluna pode ter o valor `Admin` ou`Eng`. A `Staff-type` coluna pode ter o valor `Part-time` ou`Full-time`. A `Location` coluna pode ter o valor `Office1` ou`Office2`. Os buckets do Amazon S3 para sua saída de trabalho são parecidos com os seguintes.

```
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
```

Automatizando a execução de trabalhos com um cronograma

Você pode executar novamente os DataBrew trabalhos a qualquer momento e também automatizar as execuções de trabalhos com DataBrew um cronograma.

Para executar novamente um trabalho DataBrew

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console em https://console.aws.amazon.com/databrew/.
2. No painel de navegação, escolha Trabalhos. Escolha o trabalho que você deseja executar e, em seguida, escolha Executar trabalho.

Para executar um DataBrew trabalho em um determinado horário ou de forma recorrente, crie um cronograma de DataBrew trabalho. Em seguida, você pode configurar seu trabalho para ser executado de acordo com o cronograma.

Para criar um cronograma de DataBrew trabalho

1. No painel de navegação do DataBrew console, escolha Trabalhos. Escolha a guia Programações e escolha Adicionar agenda.
2. Insira um nome para sua agenda e, em seguida, escolha um valor para a frequência de execução:
   - Recorrente — escolha com que frequência você deseja que o trabalho seja executado (por exemplo, a cada 12 horas). Em seguida, escolha em qual dia ou dias executar o trabalho. Opcionalmente, você pode inserir a hora do dia em que o trabalho é executado.
   - Em um horário específico — insira a hora do dia em que você deseja que o trabalho seja executado. Em seguida, escolha em qual dia ou dias executar o trabalho.
   - Inserir CRON — Defina o cronograma do trabalho inserindo uma expressão cron válida. Para obter mais informações, consulte Trabalhando com expressões cron para trabalhos de receitas.
3. Quando estiver satisfeito com as configurações, clique em Salvar.

Para associar um trabalho a um cronograma

1. No painel de navegação, escolha Trabalhos.
2. Escolha o trabalho com o qual você deseja trabalhar e, em Ações, escolha Editar. .
3. No painel Agendar trabalhos, escolha Associar agendamento. Escolha o nome da agenda que você deseja usar.
4. Quando estiver satisfeito com as configurações, clique em Salvar.

Trabalhando com expressões cron para trabalhos de receitas

Expressões cron têm seis campos obrigatórios, que são separados por um espaço em branco. A sintaxe é a seguinte.

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

Na sintaxe anterior, os seguintes valores e curingas são usados para os campos indicados.

Campos	Valores	Curingas
Minutos	0–59	, - * /
Horas	0–23	, - * /
Day-of-month	1–31	, - * ? / L W
Mês	1—12 ou JAN-DEC	, - * /
Day-of-week	1—7 ou SUN-SAT	, - * ? / L
Ano	1970–2199	, - * /

Use esses curingas da seguinte forma:

- A , (vírgula) curinga inclui valores adicionais. No `Month` campo, `JAN,FEB,MAR` inclui janeiro, fevereiro e março.
- O caractere curinga - (em hífen) especifica os intervalos. No `Day` campo, 1—15 inclui os dias 1 a 15 do mês especificado.
- O * (asterisco) curinga inclui todos os valores no campo. No `Hours` campo, * inclui cada hora.
- A / (barra) curinga especifica incrementos. No `Minutes` campo, você pode inserir **`1/10`** para especificar a cada 10 minutos, a partir do primeiro minuto da hora (por exemplo, 11, 21 e 31 minutos).
- O curinga ? (interrogação) especifica um ou outro. Por exemplo, suponha que no `Day-of-month` campo você insira 7. Se você não se importava em que dia da semana era o sétimo, você pode entrar? no `Day-of-week` campo.

- O caractere curinga L no `Day-of-week` campo `Day-of-month` ou especifica o último dia do mês ou da semana.
- O curinga W no campo `Day-of-month` especifica um dia da semana. No campo `Day-of-month`, `3W` especifica o dia mais próximo do terceiro dia da semana do mês.

Esses campos e valores têm as seguintes limitações:

- Não é possível especificar os campos `Day-of-month` e `Day-of-week` na mesma expressão cron. Se você especificar um valor em um dos campos, deverá usar um ? (ponto de interrogação) no outro.
- Expressões Cron que levam a taxas superiores a 5 minutos não são suportadas.

Ao criar uma programação, você pode usar os seguintes exemplos de strings cron.

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0	10	*	*	?	*	Corra às 10:00 (UTC) todos os dias
15	12	*	*	?	*	Executada às 12h15 (UTC) todos os dias
0	18	?	*	MON-FRI	*	Executada às 18h (UTC) de segunda a sexta
0	8	1	*	?	*	Corra às 8:00 AM

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
						(UTC) todo primeiro dia do mês
0/15	*	*	*	?	*	Executada a cada 15 minutos
0/10	*	?	*	MON-FRI	*	Executada a cada 10 minutos de segunda a sexta
0/5	8-17	?	*	MON-FRI	*	Executada a cada 5 minutos de segunda a sexta entre 8h e 17h55 (UTC)

Por exemplo, você pode usar a seguinte expressão cron para executar um trabalho todos os dias às 12:15 UTC.

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

Excluindo trabalhos e cronogramas de trabalho

Se você não precisar mais de um trabalho ou agenda de trabalho, poderá excluí-lo.

Para excluir um trabalho

1. No painel de navegação, escolha Trabalhos.
2. Escolha o trabalho que você deseja excluir e, em Ações, escolha Excluir. .

Para excluir uma agenda de trabalho

1. No painel de navegação, escolha Trabalhos e, em seguida, escolha a guia Agendas.
2. Escolha a agenda que você deseja excluir e, em Ações, escolha Excluir. .

Criando e trabalhando com AWS Glue DataBrew empregos de perfil

Os trabalhos de perfil executam uma série de avaliações em um conjunto de dados e enviam os resultados para o Amazon S3. As informações que a criação de perfil de dados coleta ajudam você a entender seu conjunto de dados e decidir que tipo de etapas de preparação de dados você pode querer executar em seus trabalhos de receita.

A maneira mais simples de executar um trabalho de perfil é usando as DataBrew configurações padrão. Você pode configurar seu trabalho de perfil antes de executá-lo para que ele retorne apenas as informações desejadas.

Use o procedimento a seguir para criar um trabalho DataBrew de perfil.

Para criar um emprego de perfil

1. Faça login no Console de gerenciamento da AWS e abra o DataBrew console em https://console.aws.amazon.com/databrew/.
2. Escolha TRABALHOS no painel de navegação, escolha a guia Perfil de trabalhos e, em seguida, escolha Criar trabalho.
3. Insira um nome para seu trabalho e, em seguida, escolha Criar um trabalho de perfil.
4. Para Job input, forneça o nome do conjunto de dados a ser perfilado.
5. (Opcional) Configure o seguinte no painel Configurações do perfil de dados:

   - Configurações no nível do conjunto de dados — configure os detalhes do seu trabalho de perfil para todas as colunas do seu conjunto de dados.

     Opcionalmente, você pode ativar a capacidade de detectar e contar linhas duplicadas no conjunto de dados. Você também pode escolher Ativar matriz de correlações e selecionar colunas para ver até que ponto os valores em várias colunas estão relacionados. Para obter detalhes sobre as estatísticas que você pode configurar no nível do conjunto de dados,

consulteEstatísticas configuráveis no nível do conjunto de dados. Você pode configurar estatísticas no DataBrew console ou usando a DataBrew API ou os AWS SDKs.

- Configurações em nível de coluna — Usando as configurações de perfil padrão, você pode selecionar as colunas a serem incluídas em seu trabalho de perfil. Use Adicionar substituição de configuração para selecionar as colunas para as quais limitar o número de estatísticas coletadas ou substituir a configuração padrão de determinadas estatísticas. Para obter detalhes sobre as estatísticas que você pode configurar no nível da coluna, consulteEstatísticas configuráveis no nível da coluna. Você pode configurar estatísticas no DataBrew console ou usando a DataBrew API ou os AWS SDKs.

  Certifique-se de que todas as substituições de configuração que você especificar se apliquem às colunas que você incluiu no seu trabalho de perfil. Se houver conflitos entre substituições diferentes que você configurou para uma coluna, a última substituição conflitante terá prioridade.

6. (Opcional) Você pode criar regras de qualidade de dados e aplicar conjuntos de regras adicionais associados a esse conjunto de dados ou remover os já aplicados. Para obter mais informações sobre validação da qualidade de dados, consulteValidando a qualidade dos dados em AWS Glue DataBrew.
7. No painel Configurações avançadas do trabalho, você pode escolher mais opções de como seu trabalho deve ser executado:

   - Número máximo de unidades — DataBrew processa trabalhos usando vários nós de computação, executados em paralelo. O número padrão de nós é 5. O número máximo de nós é 149.
   - Tempo limite do trabalho — Se um trabalho levar mais do que o número de minutos que você definiu aqui para ser executado, ele falhará com um erro de tempo limite. O valor padrão é 2.880 minutos ou 48 horas.
   - Número de novas tentativas — Se um trabalho falhar durante a execução, DataBrew pode tentar executá-lo novamente. Por padrão, o trabalho não é repetido.
   - Habilitar Amazon CloudWatch Logs para trabalho — Permite DataBrew publicar informações de diagnóstico no CloudWatch Logs. Esses registros podem ser úteis para solucionar problemas ou para obter mais detalhes sobre como o trabalho é processado.
8. Para o Cronograma Associado, você pode aplicar um DataBrew cronograma de trabalho para que seu trabalho seja executado em um horário específico ou de forma recorrente. Para obter mais informações, consulte Automatizando a execução de trabalhos com um cronograma.

9. Quando as configurações estiverem como você deseja, escolha Criar trabalho. Ou, se você quiser executar o trabalho imediatamente, escolha Criar e executar o trabalho.

Criando uma configuração de trabalho de perfil programaticamente no AWS Glue DataBrew

Nesta seção, você pode encontrar descrições das etapas e funções do trabalho de perfil que você pode usar programaticamente. Você pode usá-los a partir do AWS Command Line Interface(AWS CLI) ou usando um dos AWS SDKs.

Em um trabalho de perfil, você pode personalizar uma configuração para controlar como DataBrew avalia seu conjunto de dados. Você pode aplicar a configuração a um conjunto de dados ou aplicá-la a colunas específicas. Você pode criar a configuração ao criar um trabalho de perfil e depois atualizá-la a qualquer momento.

Uma estrutura de configuração de perfil inclui quatro partes:

- ProfileColumns seção
- DatasetStatisticsConfiguration seção
- ColumnStatisticsConfigurations seção
- EntityDetectorConfiguration seção para configuração de PII

Veja um exemplo a seguir.

```
{
    "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 seção

Na `ProfileColumns` seção da sua estrutura, defina as colunas do seu conjunto de dados que você deseja avaliar em seu trabalho de perfil. `ProfileColumns`é uma lista de seletores de coluna (`Selectors`). Você pode especificar um nome de coluna ou uma expressão regular em um seletor de coluna. Veja a seguir um exemplo.

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

Quando `ProfileColumns` é especificado, somente colunas cujos nomes correspondem a um nome ou expressão regular em `ProfileColumns` são incluídas na tarefa de perfil. Se o trabalho de perfil não suportar o tipo de dados de uma coluna selecionada, DataBrew ignora a coluna selecionada durante a execução do trabalho.

Se ProfileColumns for indefinido, o trabalho de perfil avalia todas as colunas suportadas. As colunas suportadas são colunas que contêm dados de um tipo de dados compatível: `ByteType` `ShortTypeIntegerType`,`LongType`,,`FloatType`,`DoubleType`,`String`, ou`Boolean`.

DatasetStatisticsConfiguration seção

Na `DatasetStatisticsConfiguration` seção da sua estrutura, você pode criar uma configuração para avaliações entre colunas. A configuração inclui `IncludedStatistics` `Overrides` e. Veja a seguir um exemplo.

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

Você pode selecionar as avaliações que deseja ter adicionando os nomes das avaliações a. `IncludedStatistics` Veja a seguir um exemplo.

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

Quando você especifica`IncludedStatistics`, somente as avaliações na lista são incluídas no trabalho de perfil. Se `IncludedStatistics` for indefinido, o trabalho de perfil executará todas as avaliações suportadas com configurações padrão. Você pode excluir todas as avaliações adicionando NENHUMA a. `IncludedStatistics` Veja a seguir um exemplo.

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

Estatísticas configuráveis no nível do conjunto de dados

Na `DatasetStatisticsConfiguration` seção de sua estrutura, um cargo de perfil apóia as avaliações mostradas na tabela a seguir.

Nome da estatística	Descrição	Tipos de dados compatíveis	Status padrão	Atributos do resultado do perfil	Tipo de resultado do perfil
CONTAGEM_LINHAS_DUPLICADAS	Contagem de linhas duplicadas no conjunto de dados	todas	Enable (Habilitar)	duplicadoRowsCoun	Int
CORRELAÇÃO	Coeficiente de correlação de Pearson entre duas colunas	número	Enable (Habilitar)	correlações (em cada coluna selecionada)	Objeto

Em`IncludedStatistics`, você pode substituir as configurações padrão de cada avaliação adicionando uma substituição. Cada substituição inclui o nome de uma avaliação específica e um mapa de parâmetros.

Em`DatasetStatisticsConfiguration`, um trabalho de perfil suporta a `CORRELATION` substituição. Essa substituição calcula o coeficiente de correlação de Pearson entre duas colunas de uma lista de colunas selecionadas. A configuração padrão é selecionar as 10 primeiras colunas numéricas. Você pode especificar um número de colunas ou uma lista de seletores de coluna para substituir a configuração padrão.

`CORRELATION`usa esses parâmetros:

- `columnNumber`— O número de colunas numéricas. O trabalho de perfil seleciona as primeiras n colunas do conjunto de dados. Esse valor deve ser maior que 1. Use `"ALL"` para selecionar todas as colunas numéricas.

- `columnSelectors:`— Lista de seletores de colunas. Cada seletor pode ter um nome de coluna ou uma expressão regular.

Veja a seguir um exemplo.

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

ColumnStatisticsConfigurations seção

Na `ColumnStatisticsConfigurations` seção da sua estrutura, você pode criar configurações para colunas específicas. `ColumnStatisticsConfigurations`é uma lista de `ColumnStatisticsConfiguration` configurações. Em `ColumnStatisticsConfigurationSelectors`, há uma lista de seletores de colunas e `Statistics` para a configuração de estatísticas. Veja a seguir um exemplo.

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

`Selectors`é uma lista de seletores de coluna. Da mesma forma`ProfileColumns`, você pode especificar um nome de coluna ou uma expressão regular em cada seletor de coluna. Quando você especifica`Selectors`, a configuração da coluna é aplicada às colunas que correspondem a

qualquer seletor de coluna em`Selectors`. Caso contrário, a configuração será aplicada a todas as colunas suportadas.

Em`Statistics`, você pode substituir as configurações das colunas selecionadas. Tal como acontece com`DatasetStatisticsConfiguration`, `Statistics` tem `IncludedStatistics` `Overrides` e.

Para selecionar as avaliações que você deseja, adicione os nomes das avaliações a. `IncludedStatistics`

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

Quando você especifica`IncludedStatistics`, somente as avaliações na lista são incluídas no trabalho de perfil. Caso contrário, o trabalho de perfil executará todas as avaliações suportadas com configurações padrão.

Você pode excluir todas as avaliações adicionando `NONE` a. `IncludedStatistics`

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

Em alguns casos, pode haver várias configurações diferentes `ColumnStatisticsConfigurations` `IncludedStatistics` que você pode aplicar à mesma coluna. Nesses casos, a tarefa de perfil seleciona a última configuração `ColumnStatisticsConfigurations` e a aplica `IncludedStatistics` à coluna selecionada. Uma nova configuração substitui as configurações mais antigas.

Estatísticas configuráveis no nível da coluna

Em`ColumnStatisticsConfigurations`, um trabalho de perfil apóia as avaliações mostradas na tabela a seguir.

Um tipo de dados compatível `number` nesta tabela significa que o tipo de dados do atributo é um dos seguintes: `ByteType`,`ShortType`,`IntegerType`,`LongType`,`FloatType`, ou`DoubleType`.

Nome da estatística	Descrição	Tipos de dados compatíveis	Status padrão	Atributos do resultado do perfil	Tipo de resultado do perfil
–	Nome da coluna.	todas	–	name	string
–	Tipo de dados da coluna.	todas	–	type	string
CONTAGEM_DE_VALORES_DISTINTOS	Número de valores distintos. Um valor distinto é o valor que aparece pelo menos uma vez.	number/boolean/string	Habilitado	distintoValuesCount	Int
ENTROPIA	Entropia (teoria da informação).	number/boolean/string	Habilitado	entropia	Duplo
INTERVALO INTER_QUARTIL	Varie entre o 25º por cento e o 75º por cento dos números.	número	Habilitado	Intervalo interquartil	Duplo
CURTOSE	Curtose da coluna.	número	Habilitado	curtose	Duplo
MAX	Valor máximo na coluna.	number/string comprimento	Habilitado	max	Int/Double
VALORES_MÁXIMOS	Lista dos valores máximos na coluna e suas contagens.	número	Habilitado	Valores máximos	Lista
MEAN	Valor médio dos valores na coluna.	number/string comprimento	Habilitado	mean	Duplo
MEDIAN	Mediana dos valores na coluna.	number/string comprimento	Habilitado	mediano	Duplo

Nome da estatística	Descrição	Tipos de dados compatíveis	Status padrão	Atributos do resultado do perfil	Tipo de resultado do perfil
DESVIO MEDIANO_ABSOLUTO	A mediana das diferenças absolutas entre cada ponto de dados e a mediana de uma coluna numérica.	número	Habilitado	medianaAbsoluteDeviation	Duplo
MIN	Valor mínimo na coluna.	number/string comprimento	Habilitado	min	Int/Double
VALORES_MÍNIMOS	Lista dos valores mínimos na coluna e suas contagens.	número	Habilitado	Valores mínimos	Lista
CONTAGEM_VALORES_FALTANTES	Número de valores faltantes na coluna. Cadeias de caracteres nulas e vazias são consideradas ausentes.	todas	Habilitado	desaparecidoValuesCount	Int
MODE	O valor que ocorre com mais frequência na coluna. Se vários valores aparecerem com tanta frequência, o modo é um desses valores.	number/string comprimento	Habilitado	modo	Int/Double

Nome da estatística	Descrição	Tipos de dados compatíveis	Status padrão	Atributos do resultado do perfil	Tipo de resultado do perfil
VALORES MAIS COMUNS	Lista dos valores mais comuns na coluna.	number/boolean/string	Habilitado	a maioria CommonValues	Lista
DETECÇÃO_DISCREPÂNCIA	Detecte valores discrepantes na coluna pelo algoritmo z_Score. Conte o número de valores discrepantes e extraia uma lista de amostras dos valores discrepantes detectados.	number/string comprimento	Habilitado	zScoreOutliersCount, zScoreOutliersSample	Int/List
PERCENTIS	Valores percentuais da coluna numérica (5%, 25%, 75%, 95%).	número	Habilitado	percentil5, percentil25, percentil75, percentil95	Duplo
RANGE	Intervalo de valores na coluna.	número	Habilitado	intervalo	Int/Double
ASSIMETRIA	Distorção dos valores na coluna.	número	Habilitado	assimetria	Duplo

Nome da estatística	Descrição	Tipos de dados compatíveis	Status padrão	Atributos do resultado do perfil	Tipo de resultado do perfil
DESVIO_PADRÃO	Desvio padrão amostral imparcial dos valores na coluna.	number/string comprimento	Habilitado	Desvio padrão	Duplo
SUM	Soma dos valores na coluna.	número	Habilitado	soma	Int/Double
CONTAGEM_DE_VALORES_ÚNICOS	Número de valores exclusivos. Um valor exclusivo significa que o valor aparece somente uma vez.	number/boolean/string	Habilitado	único ValuesCount	Int
DISTRIBUIÇÃO_DE_VALOR	Medida da distribuição dos valores na coluna por intervalo.	number/string comprimento	Habilitado	Distribuição de valor	Lista
VARIANCE	Variância dos valores na coluna.	número	Habilitado	variância	Duplo
DISTRIBUIÇÃO Z_SCORE_	Medida da distribuição dos valores do escore z dos pontos de dados por intervalo.	número	Habilitado	z ScoreDistribution	Lista
CONTAGEM_ZEROS	Número de zeros (0s) na coluna.	número	Habilitado	Contagem de zeros	Int

Em`IncludedStatistics`, você pode substituir os parâmetros padrão de cada avaliação adicionando uma substituição. Cada substituição inclui o nome de uma avaliação específica e um mapa de parâmetros.

Parâmetros para ColumnStatisticsConfigurations colunas

Em`ColumnStatisticsConfigurations`, um trabalho de perfil oferece suporte aos seguintes parâmetros.

Em alguns casos, pode haver várias configurações diferentes `ColumnStatisticsConfigurations` `IncludedStatistics` que você pode aplicar à mesma coluna. Nesses casos, a tarefa de perfil seleciona a última configuração `ColumnStatisticsConfigurations` e a aplica `IncludedStatistics` à coluna selecionada. Uma nova configuração substitui as configurações mais antigas.

VALORES_MÁXIMOS

Lista os valores máximos na coluna numérica e suas contagens. O tamanho padrão da lista é 5. Você pode substituir o tamanho da lista especificando um valor para. `sampleSize`

Configurações

`sampleSize`— O tamanho da lista que inclui o número máximo e a contagem de valores na coluna numérica. Esse valor deve ser maior que 0. Use `"ALL"` para listar todos os valores.

Exemplo

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

VALORES_MÍNIMOS

Lista os valores mínimos na coluna numérica e suas contagens. O tamanho padrão da lista é 5. Você pode substituir o tamanho da lista especificando um valor para. `sampleSize`

Configurações

`sampleSize`— O tamanho da lista que inclui o número máximo e a contagem de valores na coluna numérica. Esse valor deve ser maior que 0. Use `"ALL"` para listar todos os valores.

Exemplo

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

VALORES MAIS COMUNS

Lista os valores mais comuns na coluna e suas contagens. O tamanho padrão da lista é 50. Você pode substituir o tamanho da lista especificando um valor para. `sampleSize`

Configurações

`sampleSize`— O tamanho da lista que inclui o número máximo e a contagem de valores na coluna numérica. Esse valor deve ser maior que 0. Use `"ALL"` para listar todos os valores.

Exemplo

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

DETECÇÃO_DISCREPÂNCIA

Detecta valores discrepantes na coluna numérica ou na coluna de string (com base no comprimento da string) pelo algoritmo Z_Score.

Seu trabalho de perfil conta o número de valores discrepantes e gera uma lista de exemplos de valores discrepantes e suas pontuações z. A lista de amostras é ordenada pelo valor absoluto da pontuação z. O tamanho padrão da lista é 50.

O algoritmo Z_Score identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão. O limite padrão de valores discrepantes é 3.

Você pode fornecer mais um limite, um limite moderado, para obter mais informações. Seu limite moderado deve ser menor que seu limite. Esse recurso está desativado por padrão. Quando um limite moderado é especificado, seu trabalho de perfil retorna mais uma contagem,`zScoreMildOutliersCount`. Além disso, `zScoreOutliersSample` pode incluir uma amostra de valores discrepantes de limite moderados nesse caso.

Configurações

- `threshold`— O valor limite a ser usado ao detectar valores discrepantes. Esse valor deve ser maior ou igual a 0.
- `mildThreshold`— O valor limite moderado a ser usado ao detectar valores discrepantes. Esse valor deve ser maior ou igual a 0 e menor que`threshold`.
- `sampleSize`— O tamanho da lista que inclui valores discrepantes na coluna. Use `"ALL"` para listar todos os valores.

Exemplo

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

DISTRIBUIÇÃO_DE_VALOR

Mede a distribuição dos valores na coluna pelos intervalos dos valores. Uma tarefa de perfil agrupa valores de uma coluna numérica ou coluna de sequência de caracteres (com base no comprimento da sequência de caracteres) em compartimentos por intervalos numéricos e gera uma lista de compartimentos. Os compartimentos são consecutivos e o limite superior de um compartimento é o limite inferior do próximo compartimento.

Configurações

`binNumber`— Número de caixas. Esse valor deve ser maior que 0.

Exemplo

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

DISTRIBUIÇÃO Z_SCORE_

Mede a distribuição das pontuações z dos valores na coluna numérica. Uma tarefa de perfil agrupa pontuações z de valores em compartimentos por intervalos numéricos e gera uma lista de compartimentos. Os compartimentos são consecutivos e o limite superior de um compartimento é o limite inferior do próximo compartimento.

Configurações

`binNumber`— Número de caixas. Esse valor deve ser maior que 0.

Exemplo

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

EntityDetectorConfiguration seção para configuração de PII

Na `EntityDetectorConfiguration` seção de sua estrutura, você pode configurar os tipos de entidade em seu conjunto de dados que você DataBrew deseja detectar como informações de identificação pessoal (PII) para um trabalho de perfil.

EntityTypes

Você configura os tipos de entidade que DataBrew deseja detectar como PII para seu trabalho de perfil. Quando `EntityDetectorConfiguration` está indefinido, a detecção de entidades é desativada. Os seguintes tipos de entidade podem ser detectados em seu conjunto de dados:

- 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

O grupo de tipos de entidade também `USA_ALL` é suportado e inclui todos os tipos de entidade acima, exceto `PERSON_NAME` `DATE` e.

O tipo de `EntityTypes` é uma matriz de strings.

AllowedStatistics

Configure as estatísticas que podem ser executadas em colunas que contêm entidades detectadas. Se `AllowedStatistics` for indefinido, nenhuma estatística será calculada em colunas que

contenham entidades detectadas. Consulte Estatísticas configuráveis no nível da coluna para obter uma lista de valores válidos para o `AllowedStatistics` parâmetro.

O tipo de `AllowedStatistics` é uma matriz de `AllowedStatistics` objetos.

Segurança em AWS Glue DataBrew

A segurança na nuvem AWSé a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre vocêAWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem —AWSé responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem.AWS também fornece serviços que você pode usar com segurança. Third-partyauditores testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade aplicáveis AWS Glue DataBrew, consulte os AWS serviços em Escopo por Programa AWS de Conformidade .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS Glue DataBrew. Os tópicos a seguir mostram como configurar para atender DataBrew aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus DataBrew recursos.

Tópicos

Proteção de dados em AWS Glue DataBrew

DataBrew oferece vários recursos projetados para ajudar a proteger seus dados.

Tópicos

O AWSmodelo de responsabilidade compartilhada se aplica à proteção de dados no AWS Glue DataBrew. Conforme descrito neste modelo,AWSé responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte Perguntas frequentes sobre privacidade de dados . Para obter informações sobre proteção de dados na Europa, consulte o Centro de Regulamento Geral sobre a Proteção de Dados (RGPD).

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management(IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.

- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com DataBrew ou Serviços da AWS usa o console, a API ou AWS os SDKs.AWS CLI Quaisquer dados inseridos em tags ou em campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Criptografia inativa

DataBrew suporta criptografia de dados em repouso para DataBrew projetos e trabalhos. Projetos e trabalhos podem ler dados criptografados, e trabalhos podem gravar dados criptografados chamando AWS Key Management Service(AWS KMS) para gerar chaves e descriptografar dados. Você também pode usar chaves KMS para criptografar os registros de trabalhos gerados pelos DataBrew trabalhos. Você pode especificar chaves de criptografia usando o DataBrew console ou a DataBrew API.

> **Important**
>
> AWS Glue DataBrew suporta somente chaves AWS KMS simétricas. Para obter mais informações, consulte Chaves AWS KMS no Guia do AWS Key Management Service desenvolvedor.

Ao criar trabalhos DataBrew com a criptografia ativada, você pode usar o DataBrew console para especificar chaves de criptografia do S3-managed lado do servidor (SSE-S3) ou chaves KMS armazenadas em AWS KMS(SSE-KMS) para criptografar dados em repouso.

> **Important**
>
> Quando você usa um conjunto de dados do Amazon Redshift, os objetos descarregados no diretório temporário fornecido são criptografados com. SSE-S3

Criptografando dados gravados por trabalhos DataBrew

DataBrew os trabalhos podem gravar em destinos criptografados do Amazon S3 e Amazon CloudWatch Logs criptografados.

Tópicos

- Configurando DataBrew para usar criptografia
- Criando uma rota para AWS KMS para trabalhos de VPC
- Configurando a criptografia com AWS Chaves do KMS

Configurando DataBrew para usar criptografia

Siga este procedimento para configurar seu DataBrew ambiente para usar criptografia.

Para configurar seu DataBrew ambiente para usar criptografia

1. Crie ou atualize suas chaves do AWS KMS para dar AWS KMS permissões às funções AWS Identity and Access Management(IAM) que são passadas para os DataBrew trabalhos. Essas funções do IAM são usadas para criptografar CloudWatch registros e destinos do Amazon S3. Para obter mais informações, consulte Criptografar dados de log em CloudWatch registros usando AWS KMS o Guia do usuário do Amazon CloudWatch Logs.

   No exemplo a seguir,, *"role1""role2"*, e *"role3"* são funções do IAM que são passadas para DataBrew trabalhos. Esta declaração de política descreve uma política de chave do KMS que dá permissão às funções listadas do IAM para criptografar e descriptografar com essa chave do KMS.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logs.region.amazonaws.com",
```

```
            "AWS": [
                "role1",
                "role2",
                "role3"
            ]
        },
        "Action": [
            "kms:Encrypt*",
            "kms:Decrypt*",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:Describe*"
        ],
        "Resource": "*"
    }
```

A `Service` declaração, mostrada como`"Service": "logs.region.amazonaws.com"`, é obrigatória se você usar a chave para criptografar CloudWatch registros.

2. Certifique-se de que a AWS KMS chave esteja configurada para `ENABLED` antes de ser usada.

Para obter mais informações sobre como especificar permissões usando políticas de AWS KMS chaves, consulte Usando políticas de chaves em AWS KMS.

Criando uma rota para AWS KMS para trabalhos de VPC

Você pode se conectar diretamente ao AWS KMS através de um endpoint privado na sua VPC (virtual private cloud) em vez de fazer a conexão pela Internet. Quando você usa um VPC endpoint, a comunicação entre sua VPC e VPC AWS KMSé conduzida inteiramente dentro da rede.AWS

Você pode criar um AWS KMS VPC endpoint dentro de uma VPC. Sem essa etapa, seus DataBrew trabalhos podem falhar com um`kms timeout`. Para obter instruções detalhadas, consulte Conexão AWS KMS por meio de um VPC Endpoint no Guia do AWS Key Management Service desenvolvedor.

Ao seguir essas instruções, no console da VPC, faça o seguinte:

- Escolha Ativar nome DNS privado.
- Em Grupo de segurança, escolha o grupo de segurança (incluindo uma regra de autorreferência) que você usa para seu DataBrew trabalho que acessa o Java Database Connectivity (JDBC).

Quando você executa um DataBrew trabalho que acessa os armazenamentos de dados do JDBC, DataBrew deve ter uma rota para o endpoint.AWS KMS Você pode fornecer a rota com um gateway de conversão de endereços de rede (NAT) ou com um AWS KMS VPC endpoint. Para criar um gateway NAT, consulte Gateways NAT no Manual do usuário da Amazon VPC.

Configurando a criptografia com AWS Chaves do KMS

Quando você ativa a criptografia em um trabalho, ela se aplica tanto ao Amazon S3 quanto ao. CloudWatch A função do IAM que é passada deve ter as seguintes AWS KMS permissões.

Para obter mais informações, consulte os tópicos a seguir no Guia do usuário do Amazon Simple Storage Service:

- Para obter informações sobre isso`SSE-S3`, consulte Proteção de dados usando Server-Side criptografia com chaves de S3-Managed criptografia da Amazon (SSE-S3).
- Para obter informações sobre isso`SSE-KMS`, consulte Proteção de dados usando Server-Side criptografia com chaves AWS gerenciadas pelo KMS (). SSE-KMS

Criptografia em trânsito

AWS fornece criptografia Secure Sockets Layer (SSL) para dados em trânsito.

DataBrew O suporte para fontes de dados JDBC chega.AWS Glue Ao se conectar às fontes de dados JDBC, DataBrew usa as configurações da sua AWS Glue conexão, incluindo a opção Exigir conexão SSL. Para obter mais informações, consulte Propriedades de AWS Glue conexão -AWS Glue no Guia do AWS Glue desenvolvedor.

AWS KMS fornece criptografia “traga sua própria chave” e criptografia do lado do servidor para processamento de DataBrew extração, transformação, carregamento (ETL) e para o.AWS Glue Data Catalog

Gerenciamento de chaves

Você pode usar o IAM com DataBrew para definir usuários,AWS recursos, grupos, funções e políticas refinadas em relação a acesso, negação e muito mais.

Você pode definir o acesso aos metadados usando políticas baseadas em recursos e baseadas em identidade, dependendo das necessidades da sua organização. Resource-based as políticas listam os principais que têm acesso permitido ou negado aos seus recursos, permitindo que você configure

políticas como acesso entre contas. As políticas baseadas em identidade são anexadas de modo exclusivo a usuários, grupos e funções no IAM.

DataBrew suporta a criação de sua própria criptografia AWS KMS key“traga sua própria chave”. DataBrew também fornece criptografia do lado do servidor usando chaves KMS de for jobs.AWS KMS DataBrew

Identificação e tratamento de informações de identificação pessoal (PII)

Ao criar funções analíticas ou modelos de aprendizado de máquina, você precisa de proteções para evitar a exposição de dados de informações de identificação pessoal (PII). PII são dados pessoais que podem ser usados para identificar um indivíduo, como endereço, número de conta bancária ou número de telefone. Por exemplo, quando analistas e cientistas de dados usam conjuntos de dados para descobrir informações demográficas gerais, eles não devem ter acesso às PII de indivíduos específicos.

DataBrew fornece mecanismos de mascaramento de dados para ofuscar dados de PII durante o processo de preparação de dados. Dependendo das necessidades da sua organização, existem diferentes mecanismos de redação de dados de PII disponíveis. Você pode ofuscar os dados de PII para que os usuários não possam revertê-los ou pode tornar a ofuscação reversível.

Identificar e mascarar dados de PII DataBrew envolve a criação de um conjunto de transformações que os clientes podem usar para redigir dados de PII. Parte desse processo é fornecer estatísticas e detecção de dados de PII no painel de visão geral do Perfil de Dados no DataBrew console.

Você pode usar as seguintes técnicas de mascaramento de dados:

- Substituição — substitua os dados de PII por outros valores com aparência autêntica.
- Embaralhamento — Embaralhe o valor da mesma coluna em linhas diferentes.
- Criptografia determinística: aplique algoritmos de criptografia determinística aos valores da coluna. A criptografia determinística sempre produz o mesmo texto cifrado para um valor.
- Criptografia probabilística: aplique algoritmos de criptografia probabilística aos valores da coluna. A criptografia probabilística produz texto cifrado diferente toda vez que é aplicada.
- Descriptografia — Descriptografe colunas com base em chaves de criptografia.
- Anulação ou exclusão — substitua um determinado campo por um valor nulo ou exclua a coluna.
- Mascaramento — Use a codificação de caracteres ou mascare certas partes nas colunas.
- Hash: aplique funções de hash aos valores da coluna.

Para obter mais informações sobre o uso de transformações, consulte as etapas da receita de informações de identificação pessoal (PII). Para obter mais informações sobre o uso de trabalhos de perfil para detectar PII, incluindo uma lista dos tipos de entidade que podem ser detectados, consulte a EntityDetectorConfiguration seção sobre configuração de PII em Criando uma configuração de trabalho de perfil programaticamente.

DataBrew dependência de outros AWS serviços

Para trabalhar com o DataBrew console, você precisa de um conjunto mínimo de permissões para trabalhar com os DataBrew recursos da sua AWS conta. Além dessas DataBrew permissões, o console exige permissões dos seguintes serviços:

- CloudWatch Permissões de registros para exibir registros.
- Permissões do IAM para listar e transmitir funções.
- Permissões do Amazon EC2 para listar VPCs, sub-redes, grupos de segurança, instâncias e outros objetos. DataBrew usa essas permissões para configurar itens do Amazon EC2, como VPCs, ao executar trabalhos. DataBrew
- Permissões do Amazon S3 para listar buckets e objetos.
- AWS Glue permissões para ler objetos do AWS Glue esquema, como bancos de dados, partições, tabelas e conexões.
- AWS Lake Formation permissões para trabalhar com os data lakes do Lake Formation.

Gerenciamento de identidade e acesso para AWS Glue DataBrew

AWS Identity and Access Management(IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar DataBrew os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Autenticação com identidades
- Gerenciar o acesso usando políticas
- AWS Glue DataBrew and AWS Lake Formation
- Como AWS Glue DataBrew funciona com o IAM

- Identity-based exemplos de políticas para AWS Glue DataBrew
- AWS políticas gerenciadas para AWS Glue DataBrew
- Solução de problemas de identidade e acesso no AWS Glue DataBrew

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM.Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM(IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS.

Para acesso programático,AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário-raiz

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Usuários e grupos

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso.AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Identity-based políticas

Identity-based políticas são documentos de políticas de permissões JSON que você anexa a uma identidade (usuário, grupo ou função). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Identity-based as políticas podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Resource-based políticas

Resource-based políticas são documentos de política JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Resource-based políticas são políticas embutidas que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

DataBrew não oferece suporte a políticas baseadas em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs.AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

DataBrew não oferece suporte a ACLs.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de Controle de Serviços (SCPs): as SCPs especificam o número máximo de permissões para uma organização ou uma unidade organizacional no AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations.
- Políticas de controle de recursos (RCPs): definem o número máximo de permissões disponíveis para recursos em suas contas. Consulte mais informações em Resource control policies (RCPs) no Guia do usuário do AWS Organizations.

- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

AWS Glue DataBrew and AWS Lake Formation

AWS Glue DataBrew suporta AWS Lake Formation permissões para AWS Glue Data Catalog tabelas. Quando um conjunto de dados usa uma AWS Glue Data Catalog tabela registrada no Lake Formation, o papel do IAM fornecido aos projetos ou trabalhos deve ter as permissões DESCRIBE e SELECT Lake Formation na tabela.

AWS Glue DataBrew suporta gravação em AWS Glue Data Catalog tabelas com base em AWS Lake Formation. Quando um DataBrew trabalho usa um catálogo de dados registrado no Lake Formation, a função do IAM fornecida aos trabalhos deve ter as permissões INSERT, ALTER e DELETE do Lake Formation para as tabelas envolvidas. A função do IAM deve ter `glue:UpdateTable` permissões e também permissões para o local de dados associado à tabela do catálogo de dados.

Como AWS Glue DataBrew funciona com o IAM

Antes de usar o IAM para gerenciar o acesso DataBrew, você deve entender quais recursos do IAM estão disponíveis para uso DataBrew. Para ter uma visão de alto nível de como DataBrew e outros AWS serviços funcionam com o IAM, consulte AWS Serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- DataBrew políticas baseadas em identidade
- Resource-based políticas em DataBrew
- DataBrew Funções do IAM

DataBrew políticas baseadas em identidade

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados e as condições sob as quais as ações são permitidas ou negadas. O DataBrew é compatível com ações, recursos e chaves de condição específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, uma política AWS JSON pode especificar qual principal pode realizar ações em quais recursos e sob quais condições.

O elemento Ação de uma política JSON descreve as ações às quais você pode permitir ou negar acesso em uma política. As ações de políticas geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações políticas DataBrew usam o seguinte prefixo antes da ação:`databrew:`. Por exemplo, para conceder permissão a alguém para executar uma instância do Amazon EC2 com a operação da API `RunInstances` do Amazon EC2, inclua a ação `ec2:RunInstances` na política da pessoa. As declarações de política devem incluir um `NotAction` elemento `Action` ou. DataBrew define seu próprio conjunto de ações que descrevem as tarefas que você pode realizar com ele.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme a seguir.

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

Você também pode especificar várias ações utilizando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a ação a seguir:

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

Para ver uma lista de DataBrew ações, consulte Ações definidas por AWS Glue DataBrew no Guia do usuário do IAM.

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

A seguir estão as DataBrew APIs que não oferecem suporte a permissões em nível de recurso:

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

O recurso do DataBrew conjunto de dados tem o seguinte nome de recurso da Amazon (ARN).

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

Para obter mais informações sobre o formato dos ARNs, consulte Amazon Resource Names (ARNs) e AWS Service Namespaces.

Por exemplo, para especificar a instância de `i-1234567890abcdef0` em sua instrução, use o ARN a seguir.

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

Para especificar todas as instâncias que pertencem a uma conta específica, use o caractere curinga (*).

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

Você não pode realizar algumas DataBrew ações, como aquelas para criar recursos, em um recurso específico. Nesses casos, é necessário utilizar o caractere curinga (*).

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

Para ver uma lista dos tipos de DataBrew recursos e seus ARNs, consulte Resources Defined by AWS Glue DataBrew no Guia do usuário do IAM. Para saber com quais ações é possível especificar o ARN de cada recurso, consulte Ações definidas pelo AWS Glue DataBrew.

Chaves de condição

DataBrew não fornece nenhuma chave de condição específica do serviço, mas oferece suporte ao uso de algumas chaves de condição globais. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Exemplos

Para ver exemplos de políticas DataBrew baseadas em identidade, consulte. Identity-based exemplos de políticas para AWS Glue DataBrew

Resource-based políticas em DataBrew

DataBrew não oferece suporte a políticas baseadas em recursos.

DataBrew Funções do IAM

Uma função do IAM é uma entidade dentro da sua AWS conta que tem permissões específicas.

Usando credenciais temporárias com DataBrew

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

DataBrew suporta o uso de credenciais temporárias.

Service-linked funções

Service-linked as funções permitem que os AWS serviços acessem recursos em outros serviços para concluir uma ação em seu nome. Service-linked as funções aparecem na sua conta do IAM e são de

propriedade do serviço. Um administrador do pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

Escolha de uma função do IAM em DataBrew

Ao criar um recurso de conjunto de dados em DataBrew, você escolhe uma função do IAM para permitir o DataBrew acesso em seu nome. Se você já criou uma função de serviço ou uma função vinculada ao serviço, DataBrew fornece uma lista de funções para escolher. Certifique-se de escolher uma função que permita acesso de leitura a um bucket ou AWS Glue Data Catalog recurso do Amazon S3, conforme apropriado.

Identity-based exemplos de políticas para AWS Glue DataBrew

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do DataBrew . Eles também não podem realizar tarefas usando as AWS APIs Console de gerenciamento da AWSAWS CLI, ou. Um administrador deve criar as políticas do IAM que concedam aos usuários e aos perfis permissões para executar operações de API específicas nos recursos especificados que precisam. O administrador deve anexar essas políticas aos usuários ou grupos que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

Práticas recomendadas de política

Identity-based as políticas determinam se alguém pode criar, acessar ou excluir DataBrew recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e passe para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões

definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o DataBrew console

Para acessar o AWS Glue DataBrew console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os DataBrew recursos em sua AWS conta. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas exigidas, o console não funcionará conforme o esperado para usuários ou funções com essa política.

Para garantir que usuários e funções possam usar o DataBrew console, anexe também a seguinte política AWS gerenciada às entidades. Para obter mais informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

```
AWSDataBrewConsoleAccess
```

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a DataBrew API. Em vez disso, permita o acesso somente às ações que corresponderem a operação da API que você estiver tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS.

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

Gerenciando DataBrew recursos com base em tags

Você pode usar condições em sua política baseada em identidade para gerenciar DataBrew recursos com base em tags, por exemplo, para excluir, atualizar ou descrever os recursos. O exemplo a seguir mostra uma política que nega a exclusão de um projeto. No entanto, a exclusão é negada somente se o proprietário da tag do projeto tiver o valor de admin. Essa política também concede as permissões necessárias para negar essa ação no console.

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"}
         }
      }
   ]
}
```

Você pode anexar essa política aos usuários na sua conta. Se um usuário chamado richard-roe tentar excluir um DataBrew projeto, o recurso não deverá ser marcado como owner=admin ou owner=admin. Caso contrário, o usuário não terá permissão para excluir o projeto. A chave da etiqueta de condição Owner corresponde a Proprietário e Proprietário porque os nomes da chave de condição não diferenciam maiúsculas de minúsculas. Para obter mais informações, consulte IAM JSON Policy Elements: Condition (Elementos da política JSON do IAM: Condição) no Guia do usuário do IAM.

> **Note**
>
> ListDatasets, ListJobs,, ListProjects, ListRecipes ListRulesets, e ListSchedules não oferecem suporte ao controle de acesso baseado em tags.

AWS políticas gerenciadas para AWS Glue DataBrew

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas AWS gerenciadas do que escrever políticas você mesmo. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas AWS gerenciadas. Essas políticas abrangem casos de uso comuns e estão disponíveis em sua AWS conta. Para obter mais informações sobre políticas AWS gerenciadas, consulte políticas AWS gerenciadas no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Ocasionalmente, os serviços adicionam permissões adicionais a uma política AWS gerenciada para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política AWS gerenciada quando um novo recurso é lançado ou quando novas operações são disponibilizadas. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso,AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política ReadOnlyAccessAWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso,AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista

e descrições das políticas de funções de trabalho, consulte Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

DataBrew atualizações para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas DataBrew desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do DataBrew documento. A política gerenciada pode ser encontrada no console AWS do IAM em AwsGlueDataBrewFullAccessPolicy.

Alteração	Descrição	Data
AWSGlueDataBrewServiceRole— A permissão de leitura para AWS Glue foi adicionada.	Esta atualização adiciona`glue:GetCustomEntityType` . Essa permissão é necessária para executar trabalhos AWS Glue DataBrew de perfil com PII-identification ativado.	20 de março de 2024
AWSGlueDataBrewServiceRole- A permissão de leitura para AWS Glue foi adicionada.	Esta atualização adiciona`glue:BatchGetCustomEntityTypes` . Essa permissão é necessária para executar trabalhos AWS Glue DataBrew de perfil com PII-identification ativado.	9 de maio de 2022
AwsGlueDataBrewFullAccessPolicy- Permissões de leitura para Amazon Redshift-Data DescribeStatements e Amazon S3 GetLifecycleConfiguration foram adicionadas.	Essa atualização aumenta `redshift-data:DescribeStatement` o suporte à validação do seu SQL ao criar um Redshift-based conjunto de dados da Amazon. Também ajuda `s3:GetLifecycleConfiguration` a avaliar	4 de fevereiro de 2022

Alteração	Descrição	Data
	se o prefixo de bucket do Amazon S3 que você está fornecendo como diretório temporário tem ou não o ciclo de vida configurado. Além disso, essa alteração substitui as permissões “databrew: *” por uma lista explícita de permissões, incluindo todas as APIs. DataBrew	
AwsGlueDataBrewFullAccessPolicy- Read/write permissões para o AWS Secrets Manager foram adicionadas.	Essa atualização adiciona `secretsmanager:CreateSecret` e, `secretsmanager:GetSecretValue` para um segredo chamado`databrew!default` , um segredo padrão para uso com DataBrew transformações. Além disso, ele adiciona permissões aos CreateSecret segredos prefixados com `AwsGlueDataBrew-` para criar segredos a partir do DataBrew console. GenerateRandom, descrita na Referência da AWS Key Management Service API, é usada para gerar uma sequência de bytes aleatória que é criptograficamente segura.	18 de novembro de 2021

Alteração	Descrição	Data
AWSGlueDataBrewServiceRole- Read/write permissões para o AWS Secrets Manager foram adicionadas.	Esta atualização adiciona `secretsmanager:GetSecretValue` um segredo chamado`databrew!default` , um segredo padrão para uso com DataBrew transformações.	18 de novembro de 2021
AwsGlueDataBrewFullAccessPolicy- Read/write permissões para o AWS Secrets Manager foram adicionadas.	Essa atualização adiciona `secretsmanager:CreateSecret` e, `secretsmanager:GetSecretValue` para um segredo chamado`databrew!default` , um segredo padrão para uso com DataBrew transformações. Além disso, ele adiciona permissões aos CreateSecret segredos prefixados com `AwsGlueDataBrew-` para criar segredos a partir do DataBrew console. `kms:GenerateRandom` (https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html) é usado para gerar uma string de bytes aleatória que é criptograficamente segura.	18 de novembro de 2021

Alteração	Descrição	Data
AWSGlueDataBrewServiceRole- Read/write permissões para o AWS Secrets Manager foram adicionadas.	Esta atualização adiciona `secretsmanager:GetSecretValue` um segredo chamado`databrew!default` , um segredo padrão para uso com DataBrew transformações.	18 de novembro de 2021
AwsGlueDataBrewFullAccessPolicy- Permissões de leitura para bancos de dados de AWS Glue catálogos e permissões de criação para tabela de AWS Glue catálogos foram adicionadas.	Essa atualização adiciona permissões para listar bancos de dados de AWS Glue catálogo e criar novas tabelas de catálogo em um banco de dados existente como parte da configuração da saída para DataBrew trabalhos.	30 de junho de 2021
AwsGlueDataBrewFullAccessPolicy- Read/write permissões para o recurso de AppFlow conjunto de dados da Amazon foram adicionadas.	Essa atualização adiciona permissões para ler AppFlow fluxos e execuções de fluxo existentes da Amazon e para criar execuções de fluxo.	28 de abril de 2021

Alteração	Descrição	Data
AwsGlueDataBrewFullAccessPolicy- Permissões de leitura para conjuntos de dados do banco de dados foram adicionadas.	Esta atualização adiciona permissões para ler AWS Glue conexões existentes e criar novas AWS Glue conexões para uso com DataBrew. Além disso, para facilitar a experiência do console de criar novas conexões, ele permite a listagem dos recursos da Amazon VPC e dos clusters do Amazon Redshift. Também dá permissão para listar, mas não ler,AWS Secrets Manager segredos.	30 de março de 2021
DataBrew começou a rastrear alterações	DataBrew começou a rastrear as mudanças em suas políticas AWS gerenciadas.	30 de março de 2021

Solução de problemas de identidade e acesso no AWS Glue DataBrew

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com DataBrew um IAM.

Tópicos

- Não estou autorizado a realizar uma ação em DataBrew
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta para acessar meus DataBrew recursos

Não estou autorizado a realizar uma ação em DataBrew

Se isso Console de gerenciamento da AWS indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O erro de exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um projeto, mas não tem permissões `databrew:DescribeProject`.

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

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *`my-example-project`* usando a ação `databrew:`*`GetProject`*.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o DataBrew.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no DataBrew. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta para acessar meus DataBrew recursos

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), é possível usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é DataBrew compatível com esses recursos, consulteComo AWS Glue DataBrew funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Registro e monitoramento em DataBrew

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de DataBrew suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra.AWS fornece várias ferramentas para monitorar seus DataBrew recursos e responder a possíveis incidentes:

CloudWatch Alarmes da Amazon

> Usando CloudWatch os alarmes da Amazon, você assiste a uma única métrica durante um período de tempo especificado por você. Se a métrica exceder um determinado limite, uma notificação será enviada para um tópico AWS Auto Scaling ou política do Amazon SNS.

CloudWatch os alarmes não invocam ações porque estão em um estado específico. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos.

AWS CloudTrail Registros

CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço em DataBrew. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita DataBrew, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Validação de conformidade AWS Glue DataBrew

Third-party os auditores avaliam a segurança e a conformidade AWS Glue DataBrew como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWSé determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a Documentação AWS de segurança.

Resiliência em AWS Glue DataBrew

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade.AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Pois AWS Glue DataBrew, sugerimos que você configure seus trabalhos para usar uma ou mais tentativas. O número de novas tentativas de um trabalho é configurado no DataBrew console em Configurações avançadas do trabalho.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança da infraestrutura em AWS Glue DataBrew

Como parte de um serviço gerenciado,AWS Glue DataBrewé protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa chamadas de API AWS publicadas para acessar DataBrew pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos usar o TLS 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (PFS), como Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Tópicos

- Utilizar AWS Glue DataBrew com sua VPC
- Utilizar AWS Glue DataBrew com endpoints VPC

Utilizar AWS Glue DataBrew com sua VPC

Se você usa o Amazon VPC para hospedar seus AWS recursos, você pode configurar AWS Glue DataBrew para rotear o tráfego por meio de sua nuvem privada virtual (VPC) com base no serviço Amazon VPC. DataBrew faz isso provisionando primeiro uma interface de rede elástica na sub-rede especificada. DataBrew em seguida, anexa o grupo de segurança que você especifica a essa interface de rede para controlar o acesso. O grupo de segurança especificado deve ter regras de entrada e saída autorreferenciadas para todo o tráfego. Além disso, sua VPC deve ter nomes de host

DNS e resolução ativados. Para obter mais informações, consulte Como configurar uma VPC para se conectar aos armazenamentos de dados JDBC no Guia do desenvolvedor.AWS Glue

Para AWS Glue Data Catalog conjuntos de dados, as informações da VPC são configuradas quando você cria AWS Glue uma conexão no catálogo de dados. Para criar tabelas do Catálogo de Dados para essa conexão, execute um rastreador no AWS Glue console. Para obter mais informações, consulte Preenchendo o AWS Glue Data Catalog no Guia do AWS Glue desenvolvedor.

Para conjuntos de dados de banco de dados, especifique suas informações de VPC ao criar a conexão a partir do DataBrew console.

Para usar AWS Glue DataBrew com uma sub-rede VPC sem NAT, você deve ter um endpoint VPC gateway para o Amazon S3 e um endpoint VPC para a interface.AWS Glue Para obter mais informações, consulte Criar um endpoint de gateway e Interface de endpoints VPC AWS PrivateLink() na documentação da Amazon VPC. A interface elástica provisionada por DataBrew não tem um endereço IPv4 público e, portanto, não oferece suporte ao uso de um Gateway de Internet VPC.

No momento, não há suporte para endpoints de interface do Amazon S3. Se você estiver usando AWS Secrets Manager para armazenar seu segredo, precisará de uma rota para o Secrets Manager. Se você estiver usando criptografia, precisará de uma rota para AWS Key Management Service(AWS KMS).

Utilizar AWS Glue DataBrew com endpoints VPC

Se você usa a Amazon VPC para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e DataBrew provisionar um VPC endpoint. Usando esse VPC endpoint, você pode fazer DataBrew chamadas de API.

Não é necessário usar um DataBrew VPC endpoint com DataBrew sua VPC. Para obter mais informações, consulte Utilizar AWS Glue DataBrew com sua VPC.

Você pode usar AWS Glue com VPC endpoints em todas as AWS regiões que oferecem suporte a ambos e AWS Glue VPC endpoints.

Para obter mais informações, consulte um destes tópicos no Manual do usuário da Amazon VPC:

- O que é Amazon VPC?
- Criação de um endpoint de interface

Análise de configuração e vulnerabilidade em AWS Glue DataBrew

A configuração e os controles de TI são uma responsabilidade compartilhada entre vocêAWS e você, nosso cliente. Para obter mais informações, consulte o modelo de responsabilidade AWS compartilhada.

Monitoramento AWS Glue DataBrew

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de AWS Glue DataBrew suas outras AWS soluções.AWS fornece as seguintes ferramentas de monitoramento para observar DataBrew, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch rastrear o uso da CPU ou outras métricas de suas instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.
- O Amazon CloudWatch Events permite que você configure notificações automáticas para eventos específicos em DataBrew. Os eventos de DataBrew são entregues aos CloudWatch Eventos quase em tempo real. Você pode configurar CloudWatch Eventos para monitorar eventos e invocar alvos em resposta a eventos que indicam alterações em seus compartilhamentos de recursos. As alterações em um compartilhamento de recursos acionam eventos tanto para o proprietário do compartilhamento de recursos quanto para as entidades principais que receberam acesso ao compartilhamento de recursos. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Events.
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log a partir de instâncias do Amazon EC2 e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- AWS CloudTrailcaptura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta. Desse modo, ele fornece os arquivos de log para um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Tópicos

- Monitoramento DataBrew com CloudWatch registros
- Registrando chamadas de DataBrew API com AWS CloudTrail
- Utilizar AWS Notificações do usuário com AWS Glue Preparação de dados

Monitoramento DataBrew com a Amazon CloudWatch

Você pode monitorar DataBrew o uso CloudWatch, que coleta dados brutos e os processa em métricas legíveis e quase em tempo real. Essas estatísticas são mantidas por 15 meses, de maneira que você possa acessar informações históricas e ter uma perspectiva melhor de como o aplicativo web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

AWS Glue DataBrew relata as seguintes métricas no `AWS/DataBrew` namespace.

Métrica	Description
`SessionCount`	O número total de DataBrew sessões na conta do cliente Dimensões válidas: LogGroupName Estatística válida: soma Unidades: contagem

Automatização com eventos DataBrew CloudWatch

O Amazon CloudWatch Events permite que você automatize seus AWS serviços e responda automaticamente a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos dos AWS serviços são entregues aos CloudWatch Eventos quase em tempo real. Você pode escrever regras simples para indicar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra. Ações que podem ser automaticamente acionadas incluem:

- Invocação do comando de execução do Amazon EC2
- Transmitir o evento Amazon Kinesis Data Streams

- Ativando uma máquina de AWS Step Functions estado
- Notificar um tópico do Amazon SNS ou uma fila do Amazon SQS

DataBrew relata um evento para CloudWatch Eventos sempre que o estado de um recurso em sua AWS conta muda. Os eventos são emitidos com base no melhor esforço.

A seguir estão exemplos de vários eventos, mostrando vários estados de um DataBrew trabalho: `SUCCEEDED` `FAILEDTIMEOUT`,, `STOPPED` e.

```
{
  "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"
  }
}
```

Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Events.

Monitoramento DataBrew com CloudWatch registros

Você pode monitorar DataBrew trabalhos usando o CloudWatch Logs, que coleta informações detalhadas do subsistema de DataBrew trabalhos e as disponibiliza para análise. Esses registros podem ser úteis se você quiser obter informações sobre os recursos que seus trabalhos de perfil e receita estão usando, ou para fins de solução de problemas. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.

Registrando chamadas de DataBrew API com AWS CloudTrail

DataBrew é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em DataBrew. CloudTrail captura todas as chamadas de API DataBrew como eventos. As chamadas capturadas incluem chamadas do DataBrew console e chamadas de código para as operações DataBrew da API. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para. DataBrew Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita para DataBrew. Você também pode determinar o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

DataBrew Informações em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre em DataBrew, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de CloudTrail eventos com histórico de eventos no Guia AWS CloudTrail do usuário.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para DataBrew, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte o seguinte no Guia do usuário do AWS CloudTrail:

- Visão geral da criação de uma trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

Todas DataBrew as ações são registradas CloudTrail e documentadas na Referência da API. Por exemplo, chamadas para o `CreateDataset` `UpdateRecipe` e `StartJobRun` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do .
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Compreendendo as entradas do arquivo de DataBrew log

Novamente, uma CloudTrail trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateProfileJob` operação.

```
{
    "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"
}
```


Utilizar AWS Notificações do usuário com AWS Glue Preparação de dados

Você pode usar as Notificações AWS do Usuário para configurar canais de entrega e ser notificado sobre os eventos do AWS Glue Databrew. Você recebe uma notificação quando um evento corresponde a uma regra especificada. É possível receber notificações para eventos por meio de diversos canais, incluindo o e-mail, o Amazon Q Developer em aplicações de chat, notificações por chat ou notificações push do AWS Console Mobile Application. Você também pode ver as notificações na Central de notificações do console.AWS As notificações do usuário oferecem suporte à agregação, o que pode reduzir o número de notificações que você recebe durante eventos específicos.

Etapa da receita e referência da função

Nesta referência, você pode encontrar descrições das etapas e funções da receita que você pode usar programaticamente, a partir do AWS CLI ou usando um dos AWS SDKs. Em DataBrew, uma etapa de receita é uma ação que transforma seus dados brutos em um formulário pronto para ser consumido pelo seu pipeline de dados. Uma DataBrew função é um tipo especial de etapa de receita que executa um cálculo com base em parâmetros.

As categorias para transformações na interface do usuário incluem o seguinte:

- Etapas básicas da receita da coluna
  - Filtro
  - Coluna
- Etapas da receita de limpeza de dados
  - Formato
  - Limpa
  - Extract
- Etapas da receita de qualidade de dados
  - Missing (Ausente)
  - Inválido
  - Duplica
  - Valores discrepantes
- Etapas da receita de informações de identificação pessoal (PII)
  - Mascarar informações pessoais
  - Substitua as informações pessoais
  - Criptografar informações pessoais
  - Embaralhar linhas
- Etapas da receita da estrutura da coluna
  - Split
  - Mesclar
  - Criar
- Etapas da receita de formatação de colunas

  - Precisão decimal
  - Separador de milhares
  - Números abreviados
- Etapas da receita da estrutura de dados
  - Nest-Unnest
  - Pivot
  - Group (Grupo)
  - Ingressar
  - Union
- Etapas da receita da ciência de dados
  - Texto
  - Escala
  - Mapeamento
  - Codificação
- Funções
  - Funções matemáticas
  - Funções agregadas
  - Funções de texto
  - Perfis de data e hora
  - Funções de janela
  - Funções da Web
  - Outras funções

Para obter mais informações sobre como essas etapas e funções da receita são usadas em uma receita (incluindo o uso de expressões de condição), consulteDefinindo uma estrutura de receita.

As seções a seguir descrevem as etapas e funções da receita, organizadas de acordo com o que elas fazem.

Tópicos

- Etapas básicas da receita da coluna
- Etapas da receita de limpeza de dados

Etapas básicas da receita da coluna

Use essas ações básicas de receita de coluna para realizar transformações simples em seus dados.

Tópicos

- TO_BOOLEAN_COLUMN
- PARA COLUNA DUPLA
- PARA_NUMBER_COLUMN
- TO_STRING_COLUMN

ALTERAR_TIPO_DE_DADOS

Altera o tipo de dados de uma coluna existente.

Se o valor de uma coluna não puder ser convertido para o novo tipo, ele será substituído por NULL. Isso pode acontecer quando uma coluna de string é convertida em uma coluna inteira. Por exemplo, a string “123" se tornará o número inteiro 123, mas a string “ABC” não pode se tornar um número, então ela será substituída por um valor NULL.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— Novo tipo de coluna. Os seguintes tipos de dados são compatíveis:
  - byte: números inteiros assinados de 1 byte. O intervalo de números é de -128 a 127.
  - curto: números inteiros assinados de 2 bytes. O intervalo de números é de -32768 a 32767.
  - int: números inteiros assinados de 4 bytes. O intervalo de números é de -2147483648 a 2147483647.
  - long: números inteiros assinados de 8 bytes. O intervalo de números é de -9223372036854775808 a 9223372036854775807.
  - float: números de ponto flutuante de precisão única de 4 bytes.
  - double: números de ponto flutuante de precisão dupla de 8 bytes.
  - decimal: números decimais assinados com até 38 dígitos no total e 18 dígitos após o ponto decimal.
  - string: valores da cadeia de caracteres.
  - booleano: O tipo booleano tem um dos dois valores possíveis: `true` e `false` ou `yes` e `no`.
  - timestamp: valores que incluem os campos ano, mês, dia, hora, minuto e segundo.
  - data: valores que compreendem os campos ano, mês e dia.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
 }
```

DELETE

Remove uma coluna do conjunto de dados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

DUPLICADO

Cria uma nova coluna com um nome diferente, mas com todos os mesmos dados. Tanto a coluna antiga quanto a nova são mantidas no conjunto de dados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— Um nome para a coluna duplicada.

Example Exemplo

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

JSON_TO_STRUCTS

Converte uma string JSON em estruturas de tipo estático. Durante a conversão, ele detecta o esquema de cada objeto JSON e os mescla para obter o esquema mais genérico para representar toda a string JSON. O parâmetro “unNestLevel” especifica quantos níveis de objetos JSON devem ser convertidos em estruturas.

Parâmetros

- `sourceColumns`— Uma lista de colunas de origem.
- `regexColumnSelector` –Uma expressão regular para selecionar as colunas.
- `removeSourceColumn`— Um valor booleano. `true`Em caso afirmativo, remova a coluna de origem; caso contrário, mantenha-a.
- `unnestLevel`— O número de níveis a serem desaninhados.
- `conditionExpressions`— Expressões condicionais.

Example Exemplo

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

```
    }
}
```

MOVE_AFTER

Move uma coluna para a posição imediatamente após outra coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— O nome de outra coluna. A coluna especificada por `sourceColumn` será movida imediatamente após a coluna especificada por`targetColumn`.

Example Exemplo

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

MOVER_BEFORE

Move uma coluna para a posição imediatamente antes de outra coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— O nome de outra coluna. A coluna especificada por `sourceColumn` será movida imediatamente após a coluna especificada por`targetColumn`.

Example Exemplo

```
{
```

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

MOVER_PARA_END

Move uma coluna para a posição final (última coluna) no conjunto de dados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

MOVER_PARA_INDEX

Move uma coluna para uma posição especificada por um número.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetIndex`— A nova posição da coluna. As posições começam com 0 — então, por exemplo, 1 se refere à segunda coluna, 2 refere-se à terceira coluna e assim por diante.

Example Exemplo

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

MOVER_PARA_START

Move uma coluna para a posição inicial (primeira coluna) no conjunto de dados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

RENAME

Cria uma nova coluna com um nome diferente, mas com todos os mesmos dados. A coluna antiga é então removida do conjunto de dados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— Um novo nome para a coluna.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

Classifica os dados em uma ou mais colunas de um conjunto de dados em ordem crescente, decrescente ou personalizada.

Parâmetros

- `expressions`— Uma string que contém uma ou mais JSON-encoded strings representando expressões de classificação.
  - `sourceColumn`— Uma string que contém o nome de uma coluna existente.
  - `ordering`— A ordem pode ser ASCENDENTE ou DECRESCENTE.
  - `nullsOrdering`— A ordem dos nulos pode ser NULLS_TOP ou NULLS_BOTTOM para colocar valores nulos ou ausentes no início ou na parte inferior da coluna.
  - `customOrder`— Uma lista de strings que define uma ordem personalizada para a classificação de strings. Por padrão, as cadeias de caracteres são classificadas em ordem alfabética.
  - `isCustomOrderCaseSensitive` – Booleano. O valor padrão é `false`.

Example Exemplo

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

```
        }
    }
}
```

Example Exemplo de ordem de classificação personalizada

No exemplo a seguir, a string da expressão CustomOrder tem o formato de uma lista de objetos. Cada objeto descreve uma expressão de classificação para uma coluna.

```
[
  {
    "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

Altera o tipo de dados de uma coluna existente para BOOLEAN.

> **Note**
>
> Recomendamos usar a ação de receita CHANGE_DATA_TYPE em vez de TO_BOOLEAN_COLUMN.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— Um valor que deve ser`boolean`.

Example Exemplo

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

PARA COLUNA DUPLA

Altera o tipo de dados de uma coluna existente para DOUBLE.

> **Note**
>
> Recomendamos usar a ação de receita CHANGE_DATA_TYPE em vez de TO_DOUBLE_COLUMN.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— Um valor que deve ser`number`.

Example Exemplo

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

PARA_NUMBER_COLUMN

Altera o tipo de dados de uma coluna existente para NUMBER.

> **Note**
>
> Recomendamos usar a ação de receita CHANGE_DATA_TYPE em vez de TO_NUMBER_COLUMN.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— Um valor que deve ser`number`.

Example Exemplo

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

TO_STRING_COLUMN

Altera o tipo de dados de uma coluna existente para STRING.

> **Note**
>
> Recomendamos usar a ação de receita CHANGE_DATA_TYPE em vez de TO_STRING_COLUMN.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— Um valor que deve ser`string`.

Example Exemplo

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

Etapas da receita de limpeza de dados

Use essas etapas da receita de limpeza de dados para realizar transformações simples nos dados existentes.

Tópicos

- PADRÃO_DE_EXTRAÇÃO
- EXTRAIR VALOR_DE_OBRA
- REMOVER_COMBINADO
- SUBSTITUIR_ENTRE_DELIMITADORES
- SUBSTITUIR_ENTRE_POSIÇÕES
- SUBSTITUIR_TEXTO

CASO_CAPITAL

Altera cada string em uma coluna para colocar cada palavra em maiúscula. Em maiúsculas, a primeira letra de cada palavra é maiúscula e o resto da palavra é transformado em minúscula. Um exemplo é: The Quick Brown Fox pulou sobre a cerca.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

FORMATO_DATA

Retorna uma coluna na qual uma string de data é convertida em um valor formatado.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetDateFormat`— Um dos seguintes formatos de data:

- `mm/dd/yyyy`
- `mm-dd-yyyy`
- `dd month yyyy`
- `month yyyy`
- `dd month`

Example Exemplo

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

MINÚSCULAS

Altera cada corda em uma coluna para minúscula, por exemplo: a rápida raposa marrom pulou a cerca

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

MAIÚSCULA

Altera cada string em uma coluna para maiúscula, por exemplo: THE QUICK BROWN FOX JUMPED OVER THE FENCE

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

CASO_FRASE

Altera cada string em uma coluna para maiúsculas e minúsculas. No caso da frase, a primeira letra de cada frase é maiúscula e o resto da frase é transformado em minúscula. Um exemplo é: A raposa marrom rápida. Saltou para cima. A cerca

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

```
}
```

ADICIONAR_ASPAS DUPLAS

Coloca os caracteres em uma coluna com aspas duplas.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

ADICIONAR_PREFIXO

Adiciona um ou mais caracteres, concatenando-os como prefixo no início de uma coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— O caractere ou caracteres a serem colocados no início dos valores da coluna.

Example Exemplo

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

```
    }
}
```

ADICIONAR_CITAÇÕES_ÚNICAS

Coloca os caracteres em uma coluna com aspas simples.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

ADICIONAR_SUFIXO

Adiciona mais um caractere concatenando-os como sufixo no final de uma coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— O personagem ou caracteres a serem colocados no final da coluna.

Example Exemplo

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

```
        }
    }
}
```

EXTRAIR ENTRE_DELIMITADORES

Cria uma nova coluna, com base em delimitadores, a partir dos valores em uma coluna existente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.
- `startPattern`— Uma expressão regular, indicando o caractere ou caracteres que iniciam os valores delimitados.
- `endPattern`— Uma expressão regular, indicando o caractere delimitador ou caracteres que finalizam os valores delimitados.

Example Exemplo

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

EXTRAIR ENTRE_POSIÇÕES

Cria uma nova coluna, com base nas posições dos caracteres, a partir dos valores em uma coluna existente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `targetColumn`: o nome da nova coluna a ser criada.
- `startPosition`— A posição do personagem na qual realizar a extração.
- `endPosition`— A posição do personagem na qual finalizar o extrato.

Example Exemplo

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

PADRÃO_DE_EXTRAÇÃO

Cria uma nova coluna, com base em uma expressão regular, a partir dos valores em uma coluna existente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.
- `pattern`— Uma expressão regular que indica de qual caractere ou caracteres extrair e criar a nova coluna.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_PATTERN",
        "Parameters": {
            "pattern": "^....*...$",
```

```
            "sourceColumn": "last_name",
            "targetColumn": "first_and_last_few_characters"
        }
    }
}
```

EXTRAIR VALOR_DE_OBRA

Cria uma nova coluna com um valor extraído de um caminho especificado pelo usuário. Se a coluna de origem for do tipo Mapa, Matriz ou Estrutura, cada campo no caminho deverá ser escapado usando marcações invertidas (por exemplo, `nome`).

Parâmetros

- `targetColumn`— O nome da coluna de destino.
- `sourceColumn`— Nome da coluna de origem da qual o valor deve ser extraído.
- `path`— O caminho para a chave específica que o usuário deseja extrair. Se a coluna de origem for do tipo Mapa, Matriz ou Estrutura, cada campo no caminho deverá ser escapado usando marcações invertidas (por exemplo, `nome`).

  Considere o seguinte exemplo de informações do usuário:

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

  Veja a seguir exemplos dos caminhos que você forneceria, dependendo do tipo da coluna de origem:

  - Se a coluna de origem for do tipo map, o caminho para extrair o número do telefone residencial será:

    `` `user`.`phoneNumber`.`home` ``

- Se a coluna de origem for do tipo array, o caminho para extrair o segundo valor de “cidadania” será:

  `` `user`.`citizenship`[1] ``

- Se a coluna de origem for do tipo struct, o caminho para extrair o CEP será:

  `` `user`.`address`.`zipcode` ``

Example Exemplo

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

REMOVER_COMBINADO

Remove um ou mais caracteres de uma coluna, de acordo com o que o usuário especifica.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `collapseConsecutiveWhitespace`— Se`true`, substitui dois ou mais caracteres de espaço em branco por exatamente um caractere de espaço em branco.
- `removeAllPunctuation`— Se`true`, remove todos os seguintes caracteres: `. ! , ?`
- `removeAllQuotes`— Se`true`, remove todas as aspas simples e duplas.
- `removeAllWhitespace`— Se`true`, remove todos os caracteres de espaço em branco.
- `customCharacters`— Um ou mais personagens que podem ser interpretados.
- `customValue`— Um valor que pode ser usado.

- `removeCustomCharacters`— Se`true`, remove todos os caracteres especificados pelo `customCharacters` parâmetro.
- `removeCustomValue`— Se`true`, remove todos os caracteres especificados pelo `customValue` parâmetro.
- `punctuationally`— Se`true`, remove os seguintes caracteres se eles ocorrerem no início ou no final do valor: `. ! , ?`
- `antidisestablishmentarianism`— Se`true`, remove aspas simples e aspas duplas do início e do final do valor.
- `removeLeadingAndTrailingWhitespace`— Se`true`, remove todos os espaços em branco do início e do final do valor.
- `removeLetters`— Se`true`, remove todos os caracteres alfabéticos maiúsculos e minúsculos (até; até). `A Z a z`
- `removeNumbers`— Se`true`, remove todos os caracteres numéricos (`0`por meio de`9`).
- `removeSpecialCharacters`— Se`true`, remove todos os seguintes caracteres: `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`

Example Exemplos

```
{
    "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"
        }
    }
```

```
}
```

SUBSTITUIR_ENTRE_DELIMITADORES

Substitui os caracteres entre dois delimitadores pelo texto especificado pelo usuário.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `startPattern`— Caractere ou caracteres ou uma expressão regular, indicando onde a substituição deve começar.
- `endPattern`— Caractere ou caracteres ou uma expressão regular, indicando onde a substituição deve terminar.
- `value`— O caractere ou caracteres de substituição a serem substituídos.

Example Exemplo

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

SUBSTITUIR_ENTRE_POSIÇÕES

Substitui os caracteres entre duas posições pelo texto especificado pelo usuário.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `startPosition`— Um número indicando em qual posição de caractere na string a substituição deve começar.

- `endPosition`— Um número indicando em qual posição de caractere na string a substituição deve terminar.
- `value`— O caractere ou caracteres de substituição a serem substituídos.

Example Exemplo

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

SUBSTITUIR_TEXTO

Substitui uma sequência especificada de caracteres por outra.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— Caractere ou caracteres ou uma expressão regular, indicando quais caracteres devem ser substituídos na coluna de origem.
- `value`— O caractere ou caracteres de substituição a serem substituídos.

Example Exemplos

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

```
        }
    }
}
```

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

Etapas da receita de qualidade de dados

Use essas etapas de receita de qualidade de dados para preencher valores ausentes, remover dados inválidos ou remover duplicatas.

Tópicos

- FILTRO_DE_TIPO DE DADOS AVANÇADO
- ADVANCED_DATATYPE_FLAG
- EXCLUIR_FILHAS_DUPLICADAS
- EXTRAIR DETALHES AVANÇADOS DO TIPO DE DADOS
- PREENCHIMENTO COM MÉDIA
- PREENCHA COM_CUSTOMIZADO
- PREENCHER_COM_VAZIO
- PREENCHA COM O ÚLTIMO VALOR_VÁLIDO
- PREENCHA COM MEDIANA
- PREENCHA COM O MODO
- PREENCHA COM O MAIS FREQUENTE
- PREENCHER_COM_NULO
- PREENCHER_COM_SOMA
- BANDEIRAS_DUPLICADAS

- BANDEIRAS_DUPLICATES_IN_COLUMN
- GET_ADVANCED_DATATYPE
- REMOVER_DUPLICATAS
- REMOVER_INVÁLIDO
- REMOVER_AUSENTE
- SUBSTITUIR_POR_MÉDIA
- SUBSTITUIR_POR_PERSONALIZADO
- SUBSTITUIR_POR_VAZIO
- SUBSTITUIR_COM_LAST_VALID
- SUBSTITUIR_POR_MEDIANA
- SUBSTITUIR_COM_MODO
- SUBSTITUIR_POR_MAIS_FREQÜENTE
- SUBSTITUIR_COM_NULL
- SUBSTITUIR_POR_ROLLING_AVERAGE
- SUBSTITUIR_COM_ROLLING_SUM
- SUBSTITUIR_POR_SOMA

FILTRO_DE_TIPO DE DADOS AVANÇADO

Filtra a coluna de origem atual com base na detecção avançada do tipo de dados. Por exemplo, dada uma coluna DataBrew identificada como contendo códigos postais, essa transformação pode filtrar a coluna com base no fuso horário. Os detalhes que você pode extrair dependem do padrão detectado, conforme descrito nas notas abaixo.

Parâmetros

- `sourceColumn`— O nome de uma coluna de origem de string.
- `pattern`— O padrão a ser extraído.
- `advancedDataType`— Pode ser telefone, CEP, data e hora, estado, cartão de crédito, URL, e-mail, SSN ou sexo.
- `filter values`— Lista de valores de string com base nos quais o usuário deseja filtrar a coluna.
- `strategy`— KEEP_ROWS ou DISCARD_ROWS ou CLEAR_FILTERS ou CLEAR_OTHERS.
- `clearWithEmpty`— Booleano `true` ou`false`, para limpar linhas `empty` em vez de`null`.

Observações

- Se avançado DataType for Telefone, o padrão poderá ser AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Se avançado DataType for CEP, o padrão poderá ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.
- Se avançado DataType for Data e Hora, o padrão poderá ser DIA, MÊS, NOME DO MÊS, SEMANA, TRIMESTRE ou ANO.
- Se advanced DataType for State, o padrão poderá ser TIME_ZONE.
- Se avançado DataType for cartão de crédito, o padrão poderá ser COMPRIMENTO ou REDE.
- Se avançado DataType for URL, o padrão poderá ser PROTOCOL, TLD ou DOMAIN.

Example Exemplo

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

ADVANCED_DATATYPE_FLAG

Cria uma nova coluna de sinalização com base nos valores da coluna de origem atual. Por exemplo, dada uma coluna de origem contendo códigos postais, essa transformação pode ser usada para sinalizar valores como `true` ou `false` com base em um fuso horário específico. Os detalhes que você pode extrair dependem do padrão detectado, conforme descrito nas notas abaixo.

Parâmetros

- `sourceColumn`— O nome de uma coluna de origem de string.
- `pattern`— O padrão a ser extraído.

- `targetColumn`— O nome da coluna de destino.
- `advancedDataType`— Pode ser telefone, CEP, data e hora, estado, cartão de crédito, URL, e-mail, SSN ou sexo.
- `filter values`— Lista de valores de string com base nos quais o usuário deseja filtrar a coluna.
- `trueString`— O `true` valor da coluna de destino.
- `falseString`— O `false` valor da coluna de destino.

Observações

- Se avançado DataType for Telefone, o padrão poderá ser AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Se avançado DataType for CEP, o padrão poderá ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.
- Se avançado DataType for Data e Hora, o padrão poderá ser DIA, MÊS, NOME DO MÊS, SEMANA, TRIMESTRE ou ANO.
- Se advanced DataType for State, o padrão poderá ser TIME_ZONE.
- Se avançado DataType for cartão de crédito, o padrão poderá ser COMPRIMENTO ou REDE.
- Se avançado DataType for URL, o padrão poderá ser PROTOCOL, TLD ou DOMAIN.

Example Exemplo

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

EXCLUIR_FILHAS_DUPLICADAS

Exclui qualquer linha que corresponda exatamente a uma linha anterior no conjunto de dados. A ocorrência inicial não é excluída porque não corresponde a uma linha anterior.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRAIR DETALHES AVANÇADOS DO TIPO DE DADOS

Extrai detalhes do tipo de dados avançado. Os detalhes que você pode extrair dependem do padrão detectado, conforme descrito nas notas abaixo.

Parâmetros

- `sourceColumn`— O nome de uma coluna de origem de string.
- `pattern`— O padrão a ser extraído.
- `targetColumn`— O nome da coluna de destino.
- `advancedDataType`— Pode ser telefone, CEP, data e hora, estado, cartão de crédito, URL, e-mail, SSN ou sexo.

Observações

- Se avançado DataType for Telefone, o padrão poderá ser AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Se avançado DataType for CEP, o padrão poderá ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.
- Se avançado DataType for Data e Hora, o padrão poderá ser DIA, MÊS, NOME DO MÊS, SEMANA, TRIMESTRE ou ANO.
- Se advanced DataType for State, o padrão poderá ser TIME_ZONE.
- Se avançado DataType for cartão de crédito, o padrão poderá ser COMPRIMENTO ou REDE.

- Se avançado DataType for URL, o padrão poderá ser PROTOCOL, TLD ou DOMAIN.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

PREENCHIMENTO COM MÉDIA

Retorna uma coluna com dados ausentes substituídos pela média de todos os valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

PREENCHA COM_CUSTOMIZADO

Retorna uma coluna com dados ausentes substituídos por um valor específico.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser `datenumber`,`boolean`,`unsupported`,`string`, `outimestamp`.
- `value`— O valor personalizado a ser preenchido. O tipo de dados deve corresponder ao valor que você escolher`columnDataType`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

PREENCHER_COM_VAZIO

Retorna uma coluna com dados ausentes substituída por uma string vazia.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
```

```
}
```

PREENCHA COM O ÚLTIMO VALOR_VÁLIDO

Retorna uma coluna com dados ausentes substituídos pelo valor válido mais recente dessa coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser `datenumber`,`boolean`,`unsupported`,`string`, `outimestamp`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

PREENCHA COM MEDIANA

Retorna uma coluna com dados ausentes substituídos pela mediana de todos os valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
```

```
        }
    }
}
```

PREENCHA COM O MODO

Retorna uma coluna com dados ausentes substituída pelo modo de todos os valores.

Você também pode especificar a lógica de desempate, em que alguns dos valores são idênticos. Por exemplo, considere os seguintes valores:

`1 2 2 3 3 4`

A `modeType` of `MINIMUM` faz com `FILL_WITH_MODE` que retorne 2 como o valor do modo. Se `modeType` for`MAXIMUM`, o modo é 3. Para `AVERAGE`, o modo é 2,5.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `modeType`: como resolver valores de empate nos dados. Esse valor deve ser `MINIMUMNONE`,`AVERAGE`, ou`MAXIMUM`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
    }
}
```

PREENCHA COM O MAIS FREQUENTE

Retorna uma coluna com dados ausentes substituídos pelo valor mais frequente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
    }
}
```

PREENCHER_COM_NULO

Retorna uma coluna com valores de dados substituídos por null.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
    }
}
```

PREENCHER_COM_SOMA

Retorna uma coluna com dados ausentes substituídos pela soma de todos os valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

BANDEIRAS_DUPLICADAS

Retorna uma nova coluna com um valor especificado em cada linha que indica se essa linha corresponde exatamente a uma linha anterior no conjunto de dados. Quando as correspondências são encontradas, elas são marcadas como duplicatas. A ocorrência inicial não é marcada porque não corresponde a uma linha anterior.

Parâmetros

- `trueString`: valor a ser inserido se a linha corresponder a uma linha anterior.
- `falseString`: valor a ser inserido se a linha for exclusiva.
- `targetColumn`: nome da nova coluna inserida no conjunto de dados.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

BANDEIRAS_DUPLICATES_IN_COLUMN

Retorna uma nova coluna com um valor especificado em cada linha que indica se o valor na coluna de origem da linha corresponde a um valor em uma linha anterior da coluna de origem. Quando as

correspondências são encontradas, elas são marcadas como duplicatas. A ocorrência inicial não é marcada porque não corresponde a uma linha anterior.

Parâmetros

- `sourceColumn`: nome da coluna de origem.
- `targetColumn`: nome da coluna de destino.
- `trueString`: string a ser inserida na coluna de destino quando o valor da coluna de origem duplica um valor anterior nessa coluna.
- `falseString`: string a ser inserida na coluna de destino quando o valor da coluna de origem é diferente dos valores anteriores dessa coluna.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
}
```

GET_ADVANCED_DATATYPE

Dada uma coluna de sequência de caracteres, identifica o tipo de dados avançado da coluna, se houver.

Parâmetros

- `columnName`— O nome da coluna de caracteres.

Example Exemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

REMOVER_DUPLICATAS

Exclui uma linha inteira, se um valor duplicado for encontrado em uma coluna de origem selecionada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

REMOVER_INVÁLIDO

Exclui uma linha inteira se um valor inválido for encontrado em uma coluna dessa linha.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
}
```

REMOVER_AUSENTE

Retorna somente as linhas nas quais não faltam dados em uma coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

Example Exemplo

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

SUBSTITUIR_POR_MÉDIA

Substitui cada valor inválido em uma coluna pela média de todos os outros valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "age"
        }
    }
}
```

SUBSTITUIR_POR_PERSONALIZADO

Substitua as entidades detectadas por um valor personalizado.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `columnDataType`— O tipo de dados da coluna.
- `value`— O valor personalizado a ser usado para substituir valores inválidos.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

> **Note**
>
> Use um `sourceColumn` ou`sourceColumns`, mas não os dois.

Example Exemplo

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

```
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

SUBSTITUIR_POR_VAZIO

Substitui cada valor inválido em uma coluna por um valor vazio.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

SUBSTITUIR_COM_LAST_VALID

Substitui cada valor inválido em uma coluna pelo último valor válido.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `columnDataType`— O tipo de dados da coluna.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
    }
}
```

SUBSTITUIR_POR_MEDIANA

Substitui cada valor inválido em uma coluna pela mediana de todos os outros valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

SUBSTITUIR_COM_MODO

Substitui cada valor inválido em uma coluna pelo modo de todos os outros valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.
- `modeType`: como resolver valores de empate nos dados. Esse valor deve ser `MINIMUMNONE`,`AVERAGE`, ou`MAXIMUM`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
    }
}
```

SUBSTITUIR_POR_MAIS_FREQÜENTE

Substitui cada valor inválido em uma coluna pelo valor de coluna mais frequente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "wind_direction"
        }
    }
}
```

SUBSTITUIR_COM_NULL

Substitui cada valor inválido em uma coluna por um valor nulo.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna.
- `advancedDataType`— Tipos de dados especiais que são detectados DataBrew em uma coluna que tem o tipo de dados`string`. Os tipos que DataBrew podem ser detectados em uma `string` coluna incluem SSN, e-mail, número de telefone, sexo, cartão de crédito, URL DateTime, endereço IP, moeda ZipCode, país, região, estado e cidade.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

SUBSTITUIR_POR_ROLLING_AVERAGE

Substitui cada valor em uma coluna pela média contínua de uma “janela” anterior de linhas.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.
- `period`- — O tamanho da janela. Por exemplo, se `period` for 10, a média contínua será calculada usando as 10 linhas anteriores.

Example Exemplo

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

SUBSTITUIR_COM_ROLLING_SUM

Substitui cada valor em uma coluna pela soma contínua de uma “janela” anterior de linhas.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.
- `period`- — O tamanho da janela. Por exemplo, se `period` for 10, a soma contínua será calculada usando as 10 linhas anteriores.

Example Exemplo

```
{
```

```
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

SUBSTITUIR_POR_SOMA

Substitui cada valor inválido em uma coluna pela soma de todos os outros valores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `columnDataType`— O tipo de dados da coluna. Esse tipo deve ser`number`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

Etapas da receita de informações de identificação pessoal (PII)

Use essas etapas da receita para realizar transformações em informações de identificação pessoal (PII) em um conjunto de dados.

> **Note**
>
> Além das etapas de receita nesta seção, há etapas de DataBrew receita não projetadas especificamente para PII que você pode usar para lidar com PII. Um exemplo é DELETE uma etapa básica de receita de coluna que exclui uma coluna.

Tópicos

HASH CRIPTOGRÁFICO

Aplica um algoritmo aos valores de hash na coluna.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `secretId`: o ARN da chave secreta do Secrets Manager. A chave usada no algoritmo de prefixo do código de autenticação de mensagem baseado em hash (HMAC) para fazer o hash das colunas de origem ou `databrew!default` é a saída decodificada em base64 para o valor da chave secreta do Secrets Manager.
- `secretVersion`: opcional. O padrão é a versão mais recente do segredo.

- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.
- `createSecretIfMissing`: booleano opcional. Se verdadeiro, tentará criar o segredo em nome do chamador.
- `algorithm`: o algoritmo usado para fazer o hash de seus dados. Valores de enumeração válidos: MD5, SHA1, SHA256, SHA512, HMAC_MD5, HMAC_SHA1, HMAC_SHA256, HMAC_SHA512

  Cada opção se refere a um algoritmo de hash diferente. Essas opções com o prefixo “HMAC” se referem a um algoritmo de hash com chave e exigem o parâmetro. `secretId` Para opções sem o prefixo “HMAC”, o `secretId` parâmetro não é necessário.

  Se você não fornecer um algoritmo de hash, o serviço assumirá como padrão “HMAC_SHA256".

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "entityTypeFilter": ["USA_ALL"]
}
```

Ao trabalhar na experiência interativa, além da função do projeto, o usuário do console deve ter permissão para `secretsmanager:GetSecretValue` acessar o segredo fornecido pelo Secrets Manager.

Política de amostra:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
```

```
    ]
}
```

Você também pode optar por usar o segredo DataBrew-created padrão passando `databrew!default` como secretId e `createSecretIfMissing` o parâmetro como verdadeiro. Isso não é recomendado para produção. Qualquer pessoa com a AwsGlueDataBrewFullAccessPolicyfunção pode usar o segredo padrão.

DECIFRAR

Você pode usar a transformação DECRYPT para descriptografar dentro do. DataBrew Seus dados também podem ser descriptografados externamente DataBrew com o SDK de criptografia.AWS Se o ARN da chave KMS fornecida não corresponder ao que foi usado para criptografar a coluna, a operação de descriptografia falhará. Para obter mais informações sobre o SDK de AWS criptografia, consulte O que é o SDK de AWS criptografia no Guia do AWS Encryption SDK desenvolvedor.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `kmsKeyArn`— O ARN da chave do AWS Key Management Service a ser usada para descriptografar as colunas de origem. Para obter mais informações sobre o ARN da chave, consulte o ARN da chave no Guia do desenvolvedor.AWS Key Management Service

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
}
```

Ao trabalhar na experiência interativa, além da função do projeto, o usuário do console deve ter permissão para acessar `kms:GenerateDataKey` e usar `kms:Decrypt` a chave KMS fornecida.

Política de amostra:

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"
        ]
    }
  ]
}
```

DECIFRAR DETERMINÍSTICA

Descriptografa dados criptografados com DETERMINISTIC_ENCRYPT.

Essa transformação é autônoma se o ID secreto e a versão fornecidos não corresponderem ao que foi usado para criptografar a coluna.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `secretId`— O ARN da chave secreta do Secrets Manager a ser usada para descriptografar as colunas de origem.
- `secretVersion`: opcional. O padrão é a versão mais recente do segredo.

Exemplo

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

Ao trabalhar na experiência interativa, além da função do projeto, o usuário do console deve ter permissão para secretsmanager: GetSecretValue no segredo fornecido pelo Secrets Manager.

Política de amostra:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

ENCRIPTAÇÃO DETERMINÍSTICA

Criptografa a coluna usando AES-GCM-SIV uma chave de 256 bits. Os dados criptografados com DETERMINISTIC_ENCRYPT só podem ser descriptografados internamente com a transformação DETERMINISTIC_DECRYPT. DataBrew Essa transformação não usa o SDK AWS KMS de AWS criptografia e, em vez disso, usa a biblioteca AWS LC github.

Pode criptografar até 400 KB por célula. Não preserva o tipo de dados na descriptografia.

> **Note**
>
> Nota: Usar um segredo por mais de um ano é desencorajado.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `secretId`— O ARN da chave secreta do Secrets Manager a ser usada para criptografar as colunas de origem ou o databrew! padrão.
- `secretVersion`: opcional. O padrão é a versão mais recente do segredo.

- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.
- `createSecretIfMissing`: booleano opcional. Se verdadeiro, tentará criar o segredo em nome do chamador.

Exemplo

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123",
   "entityTypeFilter": ["USA_ALL"]
}
```

Ao trabalhar na experiência interativa, além da função do projeto, o usuário do console deve ter permissão para `secretsmanager:GetSecretValue` acessar o segredo fornecido pelo Secrets Manager.

Política de amostra

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

ENCRIPTAR

Criptografa valores nas colunas de origem com o SDK AWS de criptografia. A transformação DECRYPT pode ser usada para descriptografar dentro de. DataBrew Você também pode descriptografar os dados fora do DataBrew uso do SDK de criptografia.AWS

A transformação ENCRYPT pode criptografar até 128 MiB por célula. Ela tentará preservar o formato na decodificação. Para preservar o tipo de dado, os metadados do tipo de dado devem ser serializados para menos de 1 KB. Caso contrário, você deve definir o parâmetro `preserveDataType` como falso. Os metadados do tipo de dado serão armazenados em texto simples no contexto de criptografia. Para obter mais informações sobre o contexto de criptografia, consulte Contexto de criptografia no Guia do AWS Key Management Service desenvolvedor.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `kmsKeyArn`— O ARN da chave do AWS Key Management Service a ser usada para criptografar as colunas de origem. Para obter mais informações sobre o ARN da chave, consulte o ARN da chave no Guia do desenvolvedor.AWS Key Management Service
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.
- `preserveDataType`: booleano opcional. O valor padrão é verdadeiro. Se for falso, o tipo de dado não será armazenado.

No exemplo a seguir, `entityTypeFilter` e `preserveDataType` são opcionais.

Exemplo

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

Ao trabalhar na experiência interativa, além da função do projeto, o usuário do console deve ter permissão para `kms:GenerateDataKey` acessar a AWS KMS chave fornecida.

Política de amostra:

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

Mascara caracteres que correspondam a um valor personalizado fornecido.

Parâmetros

- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `maskSymbol`— Um símbolo que será usado para substituir caracteres especificados.
- `regex`— Se verdadeiro, trata `customValue` como um padrão regex correspondente.
- `customValue`— Todas as ocorrências (ou correspondências de regex) de `customValue` serão mascaradas na string.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.

Example Exemplo

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
```

```
        "Parameters": {
            "sourceColumns": ["company"],
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
}
```

DATA_MÁSCARA

Mascara os componentes de uma data com um símbolo de máscara especificado pelo usuário.

Parâmetros

- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `maskSymbol`— Um símbolo que será usado para substituir caracteres especificados.
- `redact`— Uma matriz de enumerações de componentes de data para mascarar. Valores de enumeração válidos: ANO, MÊS, DIA, HORA, MINUTO, SEGUNDO, MILISSEGUNDO.
- `locale`— Etiqueta de idioma IETF BCP 47 opcional. O padrão é `en`. A localidade a ser usada para formatação de data.

Example Exemplo

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

DELIMITADOR_MÁSCARA

Mascara caracteres entre dois delimitadores com um símbolo de mascaramento especificado pelo usuário.

Parâmetros

- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `maskSymbol`— Um símbolo que será usado para substituir caracteres especificados.
- `startDelimiter`— Um caractere indicando onde o mascaramento deve começar. A omissão desse parâmetro aplicará a máscara a partir do início da string.
- `endDelimiter`— Um caractere indicando onde o mascaramento deve terminar. A omissão desse parâmetro aplicará o mascaramento do StartDelimiter ao final da string.
- `preserveDelimiters`— Se verdadeiro, aplica máscara aos delimitadores.
- `alphabet`— Uma matriz de conjuntos de caracteres a serem preservados durante o mascaramento. Valores de enumeração válidos: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.

Example Exemplo

```
// 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"]
        }
    }
}
```

GAMA_DE_MÁSCARA

Mascara caracteres entre duas posições com um símbolo de mascaramento especificado pelo usuário.

Parâmetros

- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `maskSymbol`— Um símbolo que será usado para substituir caracteres especificados.
- `start`— Um número indicando em qual posição do caractere o mascaramento deve começar (indexado em 0, inclusive). A indexação negativa é permitida. A omissão desse parâmetro aplicará a máscara do início da string até 'stop'.
- `stop`— Um número indicando em qual posição do caractere o mascaramento deve terminar (indexado em 0, exclusivo). A indexação negativa é permitida. A omissão desse parâmetro aplicará a máscara do 'início' até o final da string.
- `alphabet`— Uma série de enums de conjuntos de caracteres a serem preservados durante o mascaramento. Valores de enumeração válidos: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Pode ser usada para criptografar somente as PII detectadas na coluna de texto livre.

Example Exemplo

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

SUBSTITUIR_POR_RANDOM_BETWEEN

Substitui valores por um número aleatório.

Parâmetros

- `lowerBound`— O limite inferior do intervalo de números aleatórios.
- `sourceColumns`— Uma lista de nomes de colunas existentes.

- `upperBound`— O limite superior do intervalo de números aleatórios.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
    }
}
```

SUBSTITUIR_COM_DATA_ALEATÓRIA ENTRE

Substitui valores por uma data aleatória.

Parâmetros

- `startDate`— O início do intervalo de datas a partir do qual uma data aleatória será obtida.
- `sourceColumns`— Uma lista de nomes de colunas existentes.
- `endDate`— O fim do intervalo de datas a partir do qual uma data aleatória será obtida.

Example Exemplo

```
{
    "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

Embaralha os valores em uma determinada coluna. O embaralhamento pode ocorrer com valores agrupados por uma coluna secundária.

Parâmetros

- `sourceColumns`: uma matriz de colunas existentes.
- `groupByColumns`— Uma matriz de colunas para agrupar as colunas de origem durante o embaralhamento.

Example Exemplo

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
}
```

Etapas de detecção e tratamento de discrepâncias na receita

Use essas etapas da receita para trabalhar com valores discrepantes em seus dados e realizar transformações avançadas neles.

Tópicos

FLAG_OUTLIERS

Retorna uma nova coluna contendo um valor personalizável em cada linha que indica se o valor da coluna de origem é um valor atípico.

Parâmetros

- `sourceColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `targetColumn`— Especifica o nome de uma nova coluna na qual os resultados da estratégia de avaliação discrepante devem ser inseridos.
- `outlierStrategy`— Especifica a abordagem a ser usada na detecção de valores discrepantes. Os valores válidos incluem:
  - `Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão.
  - `MODIFIED_Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da mediana em mais do que o limite médio de desvio absoluto.
  - `IQR`— identifica um valor como discrepante quando ele ultrapassa o primeiro e o último quartil dos dados da coluna. O intervalo interquartil (IQR) mede onde estão os 50% médios dos pontos de dados.
- `threshold`— Especifica o valor limite a ser usado ao detectar valores discrepantes. O `sourceColumn` valor é identificado como um valor atípico se a pontuação calculada com o `outlierStrategy` exceder esse número. O padrão é 3.
- `trueString`— Especifica o valor da string a ser usado se um valor atípico for detectado. O padrão é “True”.
- `falseString`— Especifica o valor da string a ser usado se nenhum valor atípico for detectado. O padrão é “False”.

Os exemplos a seguir exibem a sintaxe de uma única RecipeActionoperação. Uma receita contém pelo menos uma RecipeStepoperação e uma etapa de receita contém pelo menos uma ação de receita. Uma ação de receita executa a transformação de dados que você especifica. Um grupo de ações de receita é executado em ordem sequencial para criar o conjunto de dados final.

JSON

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe JSON. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em JSON

```
{
    "Action": {
        "Operation": "FLAG_OUTLIERS",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "IQR",
            "threshold": "1.5",
            "trueString": "Yes",
            "falseString": "No"
        }
    }
}
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

YAML

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe YAML. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em YAML

```
- Action:
  Operation: FLAG_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: IQR
    trueString: Outlier
    falseString: No
    threshold: '1.5'
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

REMOVE_OUTLIERS

Remove pontos de dados que são classificados como discrepantes, com base nas configurações dos parâmetros.

Parâmetros

- `sourceColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `outlierStrategy`— Especifica a abordagem a ser usada na detecção de valores discrepantes. Os valores válidos incluem:
  - `Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão.
  - `MODIFIED_Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da mediana em mais do que o limite médio de desvio absoluto.
  - `IQR`— identifica um valor como discrepante quando ele ultrapassa o primeiro e o último quartil dos dados da coluna. O intervalo interquartil (IQR) mede onde estão os 50% médios dos pontos de dados.
- `threshold`— Especifica o valor limite a ser usado ao detectar valores discrepantes. O `sourceColumn` valor é identificado como um valor atípico se a pontuação calculada com o `outlierStrategy` exceder esse número. O padrão é 3.
- `removeType`— Especifica a forma de remover os dados. Os valores válidos são `DELETE_ROWS` e `CLEAR`.
- `trimValue`— Especifica se todos ou alguns dos valores discrepantes devem ser removidos. Esse valor booleano é padronizado como. `FALSE`
  - `FALSE`— Remove todos os valores discrepantes
  - `TRUE`— Remove valores discrepantes que estão fora do limite de percentil especificado em e. `minValue` `maxValue`
- `minValue`— Indica o valor mínimo do percentil para a faixa de valores atípicos. O intervalo válido é de 0 a 100.
- `maxValue`— Indica o valor máximo do percentil para a faixa de valores atípicos. O intervalo válido é de 0 a 100.

Os exemplos a seguir exibem a sintaxe de uma única RecipeActionoperação. Uma receita contém pelo menos uma RecipeStepoperação e uma etapa de receita contém pelo menos uma ação de

receita. Uma ação de receita executa a transformação de dados que você especifica. Um grupo de ações de receita é executado em ordem sequencial para criar o conjunto de dados final.

JSON

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe JSON. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em JSON

```
{
    "Action": {
        "Operation": "REMOVE_OUTLIERS",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "outlierStrategy": "Z_SCORE",
            "threshold": "3",
            "removeType": "DELETE_ROWS",
            "trimValue": "TRUE",
            "minValue": "5",
            "maxValue": "95"
        }
    }
}
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

YAML

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe YAML. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em YAML

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    outlierStrategy: Z_SCORE
```

```
    threshold: '3'
    removeType: DELETE_ROWS
    trimValue: 'TRUE'
    minValue: '5'
    maxValue: '95'
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

REPLACE_OUTLIERS

Atualiza os valores dos pontos de dados que são classificados como discrepantes, com base nas configurações dos parâmetros.

Parâmetros

- `sourceColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `outlierStrategy`— Especifica a abordagem a ser usada na detecção de valores discrepantes. Os valores válidos incluem:
  - `Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão.
  - `MODIFIED_Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da mediana em mais do que o limite médio de desvio absoluto.
  - `IQR`— identifica um valor como discrepante quando ele ultrapassa o primeiro e o último quartil dos dados da coluna. O intervalo interquartil (IQR) mede onde estão os 50% médios dos pontos de dados.
- `threshold`— Especifica o valor limite a ser usado ao detectar valores discrepantes. O `sourceColumn` valor é identificado como um valor atípico se a pontuação calculada com o `outlierStrategy` exceder esse número. O padrão é 3.
- `replaceType`— Especifica o método a ser usado ao substituir valores discrepantes. Os valores válidos incluem:
  - `WINSORIZE_VALUES`— Especifica o uso do percentil mínimo e máximo para limitar os valores.
  - `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`— Indica o tipo de função modal a ser usada quando `replaceType` for`REPLACE_WITH_MODE`. Os valores válidos incluem o seguinte: `MINMAX`, `AVERAGE` e.
- `minValue`— Indica o valor mínimo do percentil para a faixa de valores discrepantes que deve ser aplicada quando `trimValue` usada. O intervalo válido é de 0 a 100.
- `maxValue`— Indica o valor máximo do percentil para o intervalo de valores atípicos que deve ser aplicado quando `trimValue` usado. O intervalo válido é de 0 a 100.
- `value`— Especifica o valor a ser inserido ao usar`REPLACE_WITH_CUSTOM`.
- `trimValue`— Especifica se todos ou alguns dos valores discrepantes devem ser removidos. Esse valor booleano é definido como `TRUE` when `replaceType` is `REPLACE_WITH_NULL`REPLACE_WITH_MODE, ou`WINSORIZE_VALUES`. O padrão é `FALSE` para todos os outros.
  - `FALSE`— Remove todos os valores discrepantes
  - `TRUE`—Remove valores discrepantes que estão fora do limite máximo de percentil especificado em e. `minValue` `maxValue`

Os exemplos a seguir exibem a sintaxe de uma única RecipeActionoperação. Uma receita contém pelo menos uma RecipeStepoperação e uma etapa de receita contém pelo menos uma ação de receita. Uma ação de receita executa a transformação de dados que você especifica. Um grupo de ações de receita é executado em ordem sequencial para criar o conjunto de dados final.

JSON

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe JSON. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em JSON

```
{
```

```
    "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"
        }
    }
}
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

YAML

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe YAML. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em YAML

```
- 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'
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

RESCALE_OUTLIERS_WITH_Z_SCORE

Retorna uma nova coluna com um valor atípico redimensionado em cada linha, com base nas configurações dos parâmetros. Essa ação também aplica a Z-score normalização a valores de dados em escala linear para ter uma média (μ) de 0 e desvio padrão (σ) de 1. Recomendamos essa ação para lidar com valores discrepantes.

Parâmetros

- `sourceColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `targetColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `outlierStrategy`— Especifica a abordagem a ser usada na detecção de valores discrepantes. Os valores válidos incluem:
  - `Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão.
  - `MODIFIED_Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da mediana em mais do que o limite médio de desvio absoluto.
  - `IQR`— identifica um valor como discrepante quando ele ultrapassa o primeiro e o último quartil dos dados da coluna. O intervalo interquartil (IQR) mede onde estão os 50% médios dos pontos de dados.
- `threshold`— O valor limite a ser usado ao detectar valores discrepantes. O `sourceColumn` valor é identificado como um valor atípico se a pontuação calculada com o `outlierStrategy` exceder esse número. O padrão é 3.

Os exemplos a seguir exibem a sintaxe de uma única RecipeActionoperação. Uma receita contém pelo menos uma RecipeStepoperação e uma etapa de receita contém pelo menos uma ação de receita. Uma ação de receita executa a transformação de dados que você especifica. Um grupo de ações de receita é executado em ordem sequencial para criar o conjunto de dados final.

JSON

> Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma operação de DataBrew receita, usando a sintaxe JSON. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em JSON

```
{
    "Action": {
        "Operation": "RESCALE_OUTLIERS_WITH_Z_SCORE",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "Z_SCORE",
            "threshold": "3"
        }
    }
}
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

YAML

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma operação de DataBrew receita usando a sintaxe YAML. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em YAML

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: Z_SCORE
    threshold: '3'
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

RESCALE_OUTLIERS_WITH_SKEW

Retorna uma nova coluna com um valor atípico redimensionado em cada linha, com base nas configurações dos parâmetros. Essa ação funciona para reduzir a distorção da distribuição aplicando a transformação de log ou raiz especificada. Recomendamos essa ação para lidar com dados distorcidos.

Parâmetros

- `sourceColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `targetColumn`— Especifica o nome de uma coluna numérica existente que pode conter valores discrepantes.
- `outlierStrategy`— Especifica a abordagem a ser usada na detecção de valores discrepantes. Os valores válidos incluem:
  - `Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da média em mais do que o limite de desvio padrão.
  - `MODIFIED_Z_SCORE`— Identifica um valor como um valor atípico quando ele se desvia da mediana em mais do que o limite médio de desvio absoluto.
  - `IQR`— identifica um valor como discrepante quando ele ultrapassa o primeiro e o último quartil dos dados da coluna. O intervalo interquartil (IQR) mede onde estão os 50% médios dos pontos de dados.
- `threshold`— Especifica o valor limite a ser usado ao detectar valores discrepantes. O `sourceColumn` valor é identificado como um valor atípico se a pontuação calculada com o `outlierStrategy` exceder esse número. O padrão é 3.
- `skewFunction`— Especifica o método a ser usado ao substituir valores discrepantes. Os valores válidos incluem:
  - LOG — Aplica uma forte transformação para reduzir a inclinação positiva e negativa. Esse é um logaritmo natural (2.718281828).
  - RAIZ (com`value = 3`) — Aplica uma transformação bastante forte para reduzir a inclinação positiva e negativa. (Raiz cúbica)
  - RAIZ (com`value = 2`) — Aplica uma transformação moderada para reduzir somente a inclinação positiva. (Raiz quadrada)
  - QUADRADO — Aplica uma transformação moderada para reduzir a inclinação negativa. (Quadrado)

  - Transformação personalizada — Aplica a `ROOT` transformação especificada `LOG` ou usando o número personalizado fornecido no `value` parâmetro.
- `value`— Especifica o valor a ser usado para a transformação personalizada. Se `skewFunction` for LOG, esse valor representa a base do log. Se `skewFunction` for ROOT, esse valor representa o poder da raiz.

Os exemplos a seguir exibem a sintaxe de uma única RecipeActionoperação. Uma receita contém pelo menos uma RecipeStepoperação e uma etapa de receita contém pelo menos uma ação de receita. Uma ação de receita executa a transformação de dados que você especifica. Um grupo de ações de receita é executado em ordem sequencial para criar o conjunto de dados final.

JSON

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe JSON. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em JSON

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

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

YAML

Veja a seguir um exemplo `RecipeAction` para usar como membro de um exemplo `RecipeStep` para uma DataBrew receita, usando a sintaxe YAML. Para exemplos de sintaxe que mostram uma lista de ações de receitas, consulteDefinindo uma estrutura de receita.

Example Exemplo em YAML

```
- 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'
```

Para obter mais informações sobre como usar essa ação de receita em uma operação de API, consulte CreateRecipeou UpdateRecipe. Você pode usar essas e outras operações de API em seu próprio código.

Etapas da receita da estrutura da coluna

Use essas etapas de receita de estrutura de colunas para modificar a estrutura de colunas de seus dados.

Tópicos

- OPERAÇÃO_BOOLEANA
- OPERAÇÃO_CASO
- FLAG_COLUMN_FROM_NULL
- FLAG_COLUMN_FROM_PATTERN
- MERGE
- DIVIDIR_COLUNA_ENTRE_DELIMITADOR
- DIVIDIR_COLUNA_ENTRE_POSIÇÕES
- SPLIT_COLUMN_FROM_END
- SPLIT_COLUMN_FROM_START

- DELIMITADOR MÚLTIPLO DE COLUNAS DIVIDIDAS
- DELIMITADOR SPLIT_COLUMN_SINGLE_
- SPLIT_COLUMN_WITH_INTERVALS

OPERAÇÃO_BOOLEANA

Crie uma nova coluna, com base no resultado da condição lógica IF. Retorne valor verdadeiro se a expressão booleana for verdadeira, valor falso se a expressão booleana for falsa ou retorne um valor personalizado.

Parâmetros

- `trueValueExpression`— Resultado quando a condição for atendida.
- `falseValueExpression`— Resultado quando a condição não é atendida.
- `valueExpression`— Condição booleana.
- `withExpressions`— Configuração para resultados agregados.
- `targetColumn`: um nome para a coluna recém-criada.

Você pode usar valores constantes, referências de coluna e resultados agregados em trueValueExpression, false ValueExpression e ValueExpression.

Example Exemplo: valores constantes

Valores que permanecem inalterados, como um número ou uma frase.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Exemplo: referências de coluna

Valores que são colunas no conjunto de dados.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Exemplo: resultados agregados

Valores que são calculados por funções agregadas. Uma função agregada executa um cálculo em uma coluna e retorna um único valor.

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

```
}
```

Os usuários precisam converter o JSON em uma string escapando.

Observe que os nomes dos parâmetros em true ValueExpressionValueExpression, false e valueExpression devem corresponder aos nomes em withExpressions. Para usar os resultados agregados de algumas colunas, você precisa criar parâmetros para elas e fornecer as funções agregadas.

Example Exemplo:

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Exemplo: and/or.

Você pode usar e e ou para combinar várias condições.

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

Funções agregadas válidas

A tabela abaixo mostra todas as funções agregadas válidas que podem ser usadas em uma operação booleana.

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
Numérico	Soma	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Retorna a soma de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Média	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" } ]`	Retorna a média de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Desvio médio absoluto	`:desvio absoluto médio.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Retorna o desvio médio absoluto de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Mediana	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Retorna a mediana de `column.1`
	Produto	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Retorna o produto de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Desvio padrão	`:desvio-padrão.column.1`	`[ { "name": "standard deviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Retorna o desvio padrão de `column.1`
	Variação	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Retorna a variância de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Erro padrão da média	`:erro padrão de mean.column.1`	`[ { "name": "standard errorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Retorna o erro padrão da média de `column.1`
	Distorção	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Retorna a distorção de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Curtose	`:curtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Retorna a curtose de `column.1`
Datetime/ Numeric/Text	Contagem	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Retorna o número total de linhas em `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Contagem de distintos	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Retorna o número total de linhas distintas em `column.1`
	Mín.	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Retorna o valor mínimo de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Máx	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Retorna o valor máximo de `column.1`

Condições válidas em uma ValueExpression

A tabela abaixo mostra as condições suportadas e as expressões de valor que você pode usar.

Tipo de coluna	Condição	Expressão de valor	Description
String	Contém	contém (`coluna`, 'texto')	Condição para testar se o valor na coluna contém texto
	Não contém	! contém (`coluna`, 'texto')	Condição para testar se o valor na coluna não contém texto
	Correspondências	correspondências (`coluna`, 'padrão')	Condição para testar se o valor na coluna corresponde ao padrão

Tipo de coluna	Condição	Expressão de valor	Description
	Não coincide	! correspondências (`coluna`, 'padrão')	Condição para testar se o valor na coluna não corresponde ao padrão
	Inicia com	StartsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna começa com texto
	Não começa com	! StartsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna não começa com texto
	Termina com	endsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna termina com texto
	Não termina com	! endsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna não termina com texto
Numérico	Menor que	`coluna` < número	Condição para testar se o valor na coluna é menor que o número
	Menor ou igual a	`coluna` <= número	Condição para testar se o valor na coluna é menor ou igual ao número
	Maior que	`coluna` > número	Condição para testar se o valor na coluna é maior que o número

Tipo de coluna	Condição	Expressão de valor	Description
	Maior ou igual a	`coluna` >= número	Condição para testar se o valor na coluna é maior ou igual ao número
	Está entre	isBetween (`coluna` , minNumber, maxNumber)	Condição para testar se o valor na coluna está entre minNumber e maxNumber
	Não está entre	! isBetween (`coluna` , minNumber, maxNumber)	Condição para testar se o valor na coluna não está entre minNumber e maxNumber
Booleano	É verdade	`coluna` = VERDADEIRO	Condição para testar se o valor na coluna é booleano TRUE
	É falso	`coluna` = FALSO	Condição para testar se o valor na coluna é booleano FALSE
Date/Timestamp	Antes de	`coluna` < 'data'	Condição para testar se o valor na coluna é anterior à data
	Anterior ou igual a	`coluna` <= 'data'	Condição para testar se o valor na coluna é anterior ou igual à data
	Mais tarde do que	`coluna` > 'data'	Condição para testar se o valor na coluna é posterior à data

Tipo de coluna	Condição	Expressão de valor	Description
	Mais tarde ou igual a	`coluna` >= 'data'	Condição para testar se o valor na coluna é posterior ou igual à data
String/Numeric/Date/ Timestamp	É exatamente	`coluna` = 'valor'	Condição para testar se o valor na coluna é exatamente o valor
	Não é	`coluna`! = 'valor'	Condição para testar se o valor na coluna não é valor
	Está faltando	IsMissing (`coluna`)	Condição para testar se o valor na coluna está ausente
	Não está faltando	! IsMissing (`coluna`)	Condição para testar se o valor na coluna não está ausente
	É válido	isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)
	Não é válido	! isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna não é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)

Tipo de coluna	Condição	Expressão de valor	Description
Aninhado	Está faltando	IsMissing (`coluna`)	Condição para testar se o valor na coluna está ausente
	Não está faltando	! IsMissing (`coluna`)	Condição para testar se o valor na coluna não está ausente
	É válido	isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)
	Não é válido	! isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna não é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)

OPERAÇÃO_CASO

Crie uma nova coluna, com base no resultado da condição lógica CASE. A operação de caso passa pelas condições do caso e retorna um valor quando a primeira condição é atendida. Quando uma condição é verdadeira, a operação interrompe a leitura e retorna o resultado. Se nenhuma condição for verdadeira, ele retornará o valor padrão.

Parâmetros

- `valueExpression`— Condições.
- `withExpressions`— Configuração para resultados agregados.
- `targetColumn`— Nome da coluna recém-criada.

Example Exemplo

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

Funções agregadas válidas

A tabela abaixo mostra todas as funções agregadas válidas que podem ser usadas em uma operação de caso.

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
Numérico	Soma	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Retorna a soma de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Média	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" } ]`	Retorna a média de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Desvio médio absoluto	`:desvio absoluto médio.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Retorna o desvio médio absoluto de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Mediana	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Retorna a mediana de `column.1`
	Produto	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Retorna o produto de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Desvio padrão	`:desvio-padrão.column.1`	`[ { "name": "standard deviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Retorna o desvio padrão de `column.1`
	Variação	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Retorna a variância de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Erro padrão da média	`:erro padrão de mean.column.1`	`[ { "name": "standard errorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Retorna o erro padrão da média de `column.1`
	Distorção	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Retorna a distorção de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Curtose	`:curtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Retorna a curtose de `column.1`
Datetime/ Numeric/Text	Contagem	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Retorna o número total de linhas em `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Contagem de distintos	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Retorna o número total de linhas distintas em `column.1`
	Mín.	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Retorna o valor mínimo de `column.1`

Tipo de coluna	Condição	Expressão de valor	Com expressões	Valor de retorno
	Máx	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Retorna o valor máximo de `column.1`

Condições válidas em uma ValueExpression

A tabela abaixo mostra as condições suportadas e as expressões de valor que você pode usar.

Tipo de coluna	Condição	Expressão de valor	Description
String	Contém	contém (`coluna`, 'texto')	Condição para testar se o valor na coluna contém texto
	Não contém	! contém (`coluna`, 'texto')	Condição para testar se o valor na coluna não contém texto
	Correspondências	correspondências (`coluna`, 'padrão')	Condição para testar se o valor na coluna corresponde ao padrão

Tipo de coluna	Condição	Expressão de valor	Description
	Não coincide	! correspondências (`coluna`, 'padrão')	Condição para testar se o valor na coluna não corresponde ao padrão
	Inicia com	StartsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna começa com texto
	Não começa com	! StartsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna não começa com texto
	Termina com	endsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna termina com texto
	Não termina com	! endsWith (`coluna`, 'texto')	Condição para testar se o valor na coluna não termina com texto
Numérico	Menor que	`coluna` < número	Condição para testar se o valor na coluna é menor que o número
	Menor ou igual a	`coluna` <= número	Condição para testar se o valor na coluna é menor ou igual ao número
	Maior que	`coluna` > número	Condição para testar se o valor na coluna é maior que o número

Tipo de coluna	Condição	Expressão de valor	Description
	Maior ou igual a	`coluna` >= número	Condição para testar se o valor na coluna é maior ou igual ao número
	Está entre	isBetween (`coluna` , minNumber, maxNumber)	Condição para testar se o valor na coluna está entre minNumber e maxNumber
	Não está entre	! isBetween (`coluna` , minNumber, maxNumber)	Condição para testar se o valor na coluna não está entre minNumber e maxNumber
Booleano	É verdade	`coluna` = VERDADEIRO	Condição para testar se o valor na coluna é booleano TRUE
	É falso	`coluna` = FALSO	Condição para testar se o valor na coluna é booleano FALSE
Date/Timestamp	Antes de	`coluna` < 'data'	Condição para testar se o valor na coluna é anterior à data
	Anterior ou igual a	`coluna` <= 'data'	Condição para testar se o valor na coluna é anterior ou igual à data
	Mais tarde do que	`coluna` > 'data'	Condição para testar se o valor na coluna é posterior à data

Tipo de coluna	Condição	Expressão de valor	Description
	Mais tarde ou igual a	`coluna` >= 'data'	Condição para testar se o valor na coluna é posterior ou igual à data
String/Numeric/Date/ Timestamp	É exatamente	`coluna` = 'valor'	Condição para testar se o valor na coluna é exatamente o valor
	Não é	`coluna`! = 'valor'	Condição para testar se o valor na coluna não é valor
	Está faltando	IsMissing (`coluna`)	Condição para testar se o valor na coluna está ausente
	Não está faltando	! IsMissing (`coluna`)	Condição para testar se o valor na coluna não está ausente
	É válido	isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)
	Não é válido	! isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna não é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)

Tipo de coluna	Condição	Expressão de valor	Description
Aninhado	Está faltando	IsMissing (`coluna`)	Condição para testar se o valor na coluna está ausente
	Não está faltando	! IsMissing (`coluna`)	Condição para testar se o valor na coluna não está ausente
	É válido	isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)
	Não é válido	! isValid (`coluna`, tipo de dados)	Condição para testar se o valor na coluna não é válido (o valor é do tipo de dados ou pode ser convertido em tipo de dados)

FLAG_COLUMN_FROM_NULL

Cria uma nova coluna, com base na presença de valores nulos em uma coluna existente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— O nome de uma nova coluna a ser criada.
- `flagType`— Um valor que deve ser definido como`Null values`.
- `trueString`— Um valor para a nova coluna, se um valor nulo for encontrado na fonte. Se nenhum valor for especificado, o padrão será `True`.
- `falseString`— Um valor para a nova coluna, se um valor não nulo for encontrado na fonte. Se nenhum valor for especificado, o padrão será `False`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

Cria uma nova coluna, com base na presença de um padrão especificado pelo usuário em uma coluna existente.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`— O nome de uma nova coluna a ser criada.
- `flagType`— Um valor que deve ser definido como`Pattern`.
- `pattern`— Uma expressão regular, indicando o padrão a ser avaliado.
- `trueString`— Um valor para a nova coluna, se um valor nulo for encontrado na fonte. Se nenhum valor for especificado, o padrão será `True`.
- `falseString`— Um valor para a nova coluna, se um valor não nulo for encontrado na fonte. Se nenhum valor for especificado, o padrão será `False`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
```

```
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
    }
}
```

MERGE

Mescla duas ou mais colunas em uma nova coluna.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de uma ou mais colunas a serem mescladas.
- `delimiter`— Um separador opcional entre os valores, para aparecer na coluna de destino.
- `targetColumn`— O nome da coluna mesclada a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

DIVIDIR_COLUNA_ENTRE_DELIMITADOR

Divide uma coluna em três novas colunas, de acordo com um delimitador inicial e final.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `patternOption1`— Uma JSON-encoded string representando um ou mais caracteres que indicam o primeiro delimitador.
- `patternOption2`— Uma JSON-encoded string representando um ou mais caracteres que indicam o segundo delimitador.
- `pattern`— Um ou mais caracteres para usar como separador ao dividir os dados.
- `includeInSplit`— Se verdadeiro, inclui o padrão na nova coluna; caso contrário, o padrão será descartado.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_DELIMITER",
        "Parameters": {
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

DIVIDIR_COLUNA_ENTRE_POSIÇÕES

Divide uma coluna em três novas colunas, de acordo com os deslocamentos que você especificar.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `startPosition`— A posição do personagem em que a divisão deve começar.
- `endPosition`— A posição do personagem em que a divisão deve terminar.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_POSITIONS",
```

```
        "Parameters": {
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
        }
    }
}
```

SPLIT_COLUMN_FROM_END

Divide uma coluna em duas novas colunas, em um deslocamento do final da string.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `position`— A posição do caractere, da extremidade direita da string, onde a divisão deve ocorrer.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
}
```

SPLIT_COLUMN_FROM_START

Divide uma coluna em duas novas colunas, em um deslocamento do início da string.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `position`— A posição do caractere, a partir da extremidade esquerda da string, onde a divisão deve ocorrer.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

DELIMITADOR MÚLTIPLO DE COLUNAS DIVIDIDAS

Divide uma coluna de acordo com vários delimitadores.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `patternOptions`— Uma JSON-encoded sequência de caracteres representando um ou mais padrões que determinam os critérios de divisão.
- `pattern`— Um ou mais caracteres para usar como separador ao dividir os dados.
- `limit`— Quantas divisões realizar. O mínimo é 1; o máximo é 20.
- `includeInSplit`— Se verdadeiro, inclui o padrão na nova coluna; caso contrário, o padrão será descartado.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
```

```
}
```

DELIMITADOR SPLIT_COLUMN_SINGLE_

Divide uma coluna em uma ou mais colunas novas, de acordo com um delimitador específico.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— Um ou mais caracteres para usar como separador ao dividir os dados.
- `limit`— Quantas divisões realizar. O mínimo é 1; o máximo é 20.
- `includeInSplit`— Se verdadeiro, inclui o padrão na nova coluna; caso contrário, o padrão será descartado.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_SINGLE_DELIMITER",
        "Parameters": {
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVALS

Divide uma coluna em intervalos de n caracteres, onde você especifica n.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `startPosition`— A posição do personagem em que a divisão deve começar.
- `interval`— O número de caracteres a serem ignorados antes da próxima divisão.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

Etapas da receita de formatação de colunas

Use as etapas da receita de formatação de colunas para alterar o formato dos dados em suas colunas.

Tópicos

FORMATO_NÚMERO

Retorna uma coluna na qual um valor numérico é convertido em uma string formatada.

Parâmetros

- `sourceColumn` – String. O nome de uma coluna existente.
- `decimalPlaces`— Inteiro. O valor do número de dígitos após o separador decimal.
- `numericDecimalSeparator` – String. Um dos seguintes valores indicando o separador decimal:
  - "."
  - ","
- `numericThousandSeparator` – String. Um dos seguintes valores indicando o separador de mil:
  - nulo. Indica que o separador de mil não está ativado.
  - ","

  - " "
  - "."
  - "\\"
- `numericAbbreviatedUnit` – String. Um dos seguintes valores indicando a unidade de abreviatura:
  - nulo. Indica que uma unidade de abreviatura não está ativada.
  - “MIL”
  - “MILHÃO”
  - “BILHÃO”
  - “TRILHÃO”
- `numericUnitAbbreviation` – String. Um dos valores a seguir ou qualquer valor personalizado, indicando a abreviatura da unidade:
  - nulo. Indica que a abreviatura da unidade não está ativada.
  -

Unidade de abreviatura	Opções
Milhares	K, k, M, mil, personalizado
Million	M, m, MM, milhão, personalizado
Billion	B, bilhão, bilhão, personalizado
Triliões	T, tn, trilhão, personalizado

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
```

```
            "numericUnitAbbreviation": "K"
        }
    }
}
```

FORMATAR NÚMERO DE TELEFONE

Retorna uma coluna na qual uma sequência de números de telefone é convertida em um valor formatado.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `phoneNumberFormat`: o formato para o qual converter o número de telefone. Se nenhum formato for especificado, o padrão será `E.164`, um formato de número de telefone padrão reconhecido internacionalmente. Os valores válidos incluem:
  - `E164`(omitir o período após`E`)
- `defaultRegion`: um código de região válido que consiste em duas ou três letras maiúsculas que especifica a região do número de telefone quando nenhum código de país está presente no próprio número. No máximo, uma das `defaultRegion` ou `defaultRegionColumn` pode ser fornecida.
- `defaultRegionColumn`— O nome de uma coluna do tipo de dados avançado`Country`. O código da região da coluna especificada é usado para determinar o código do país para o número de telefone quando nenhum código de país está presente no próprio número. No máximo, uma das `defaultRegion` ou `defaultRegionColumn` pode ser fornecida.

Observações

- As entradas que não podem ser formatadas para um número de telefone válido permanecem inalteradas.
- Se nenhuma região padrão for fornecida e um número de telefone não começar com o símbolo de adição (+) e o código de chamada do país, o número de telefone não será formatado.

Example

Exemplo: região padrão fixa

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

Exemplo: opção de coluna de região padrão

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

Etapas da receita da estrutura de dados

Use essas etapas da receita para tabular e resumir dados de diferentes perspectivas ou para executar funções avançadas.

Tópicos

- DO NINHO À MATRIZ
- DO NINHO AO MAPA
- NEST_TO_STRUCT
- UNNEST_ARRAY
- MAPA_DO_UNNEST
- UNNEST_STRUCT
- UNNEST_STRUCT_N
- GROUP_BY

- JOIN
- PIVOT
- SCALE
- TRANSPÕEM
- UNION
- UNPIVOT

DO NINHO À MATRIZ

Converte colunas selecionadas pelo usuário em valores de matriz. A ordem das colunas selecionadas é mantida durante a criação da matriz resultante. Os diferentes tipos de dados de coluna são convertidos em um tipo comum que suporta os tipos de dados de todas as colunas.

Parâmetros

- `sourceColumns`— Lista das colunas de origem.
- `targetColumn`— O nome da coluna de destino.
- `removeSourceColumns`— Contém o valor `true` ou indica `false` se o usuário deseja ou não remover as colunas de origem selecionadas.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

DO NINHO AO MAPA

Converte colunas selecionadas pelo usuário em pares de valores-chave, cada uma com uma chave representando o nome da coluna e um valor representando o valor da linha. A ordem da coluna

selecionada não é mantida durante a criação do mapa resultante. Os diferentes tipos de dados de coluna são convertidos em um tipo comum que suporta os tipos de dados de todas as colunas.

Parâmetros

- `sourceColumns`— Lista das colunas de origem.
- `targetColumn`— O nome da coluna de destino.
- `removeSourceColumns`— Contém o valor `true` ou indica `false` se o usuário deseja ou não remover as colunas de origem selecionadas.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_STRUCT

Converte colunas selecionadas pelo usuário em pares de valores-chave, cada uma com uma chave representando o nome da coluna e um valor representando o valor da linha. A ordem das colunas selecionadas e o tipo de dados de cada coluna são mantidos na estrutura resultante.

Parâmetros

- `sourceColumns`— Lista das colunas de origem.
- `targetColumn`— O nome da coluna de destino.
- `removeSourceColumns`— Contém o valor `true` ou indica `false` se o usuário deseja ou não remover as colunas de origem selecionadas.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

UNNEST_ARRAY

Desaninha uma coluna do tipo `array` em uma nova coluna. Se a matriz contiver mais de um valor, uma linha correspondente a cada elemento será gerada. Essa função desaninha somente um nível de uma coluna de matriz.

Parâmetros

- `sourceColumn`— O nome de uma coluna existente. Essa coluna deve ser do `struct` tipo.
- `targetColumn`— Nome da coluna de destino que é gerada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
}
```

MAPA_DO_UNNEST

Desaninha uma coluna do tipo `map` e gera uma coluna para a chave e o valor. Se houver mais de um par de valores-chave, uma linha correspondente a cada valor de chave será gerada. Essa função só desaninha um nível de uma coluna do mapa.

Parâmetros

- `sourceColumn`— O nome de uma coluna existente. Essa coluna deve ser do `struct` tipo.
- `removeSourceColumn`— Se`true`, a coluna de origem for excluída após a conclusão da função.
- `targetColumn`— Se fornecida, cada coluna gerada começará com isso como prefixo.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

Desaninha uma coluna do tipo `struct` e gera uma coluna para cada uma das chaves presentes na estrutura. Essa função só desaninha a estrutura de nível um.

Parâmetros

- `sourceColumn`— O nome de uma coluna existente. Essa coluna deve ser do tipo estrutura.
- `removeSourceColumn`— Se`true`, a coluna de origem for excluída após a conclusão da função.
- `targetColumn`— Se fornecida, cada coluna gerada começará com isso como prefixo.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
```

```
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

Cria uma nova coluna para cada campo de uma coluna do tipo selecionada`struct`.

Por exemplo, dada a seguinte estrutura:

```
            user {
               name: “Ammy”
               address: {
                  state: "CA",
                  zipcode: 12345
               }
            }
```

Essa função cria 3 colunas:

nome do usuário	user.address.state	user.address.zip code
Ammy	CA	12345

Parâmetros

- `sourceColumns`— Lista das colunas de origem.
- `regexColumnSelector`— Uma expressão regular para selecionar as colunas a serem desaninhadas.
- `removeSourceColumn`— Um valor booleano. Se verdadeiro, remova a coluna de origem; caso contrário, mantenha-a.
- `unnestLevel`— O número de níveis a serem desaninhados.
- `delimiter`— O delimitador é usado no nome da coluna recém-criada para separar os diferentes níveis da estrutura. Por exemplo: se o delimitador for “/”, o nome da coluna estará neste formato: “user/address/state”.

- `conditionExpressions`— Expressões condicionais.

Example Exemplo

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

GROUP_BY

Resume os dados agrupando linhas por uma ou mais colunas e aplicando uma função de agregação a cada grupo.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas que formam a base de cada grupo.
- `groupByAggFunctions`— Uma JSON-encoded string representando uma lista de funções de agregação a serem aplicadas. (Se você não quiser agregação, especifique`UNAGGREGATED`.)
- `useNewDataFrame`— Se verdadeiro, os resultados de GROUP_BY são disponibilizados na sessão do projeto, substituindo seu conteúdo atual.

Example Exemplo

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
```

```
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
      }
    }
  }
]
```

JOIN

Executa uma operação de junção em dois conjuntos de dados.

Parâmetros

- `joinKeys`— Uma JSON-encoded string representando uma lista de colunas de cada conjunto de dados para atuar como chaves de junção.
- `joinType`— O tipo de junção a ser realizada. Deve ser um dos seguintes: `INNER_JOIN` | `LEFT_JOIN` | `RIGHT_JOIN` | `OUTER_JOIN` | `LEFT_EXCLUDING_JOIN` | `RIGHT_EXCLUDING_JOIN` | `OUTER_EXCLUDING_JOIN`
- `leftColumns`— Uma JSON-encoded string representando uma lista de colunas do conjunto de dados ativo atual.
- `rightColumns`— Uma JSON-encoded string representando uma lista de colunas de outro conjunto de dados (secundário) para unir ao atual.
- `secondInputLocation`— Uma URL do Amazon S3 que é resolvida para o arquivo de dados do conjunto de dados secundário.
- `secondaryDatasetName`— O nome do conjunto de dados secundário.

Example Exemplo

```
{
    "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

Converte todos os valores de linha em uma coluna selecionada em colunas individuais com valores.

Parâmetros

- `sourceColumn`— O nome de uma coluna existente. A coluna pode ter no máximo 10 valores distintos.
- `valueColumn`— O nome de uma coluna existente. A coluna pode ter no máximo 10 valores distintos.
- `aggregateFunction`— O nome de uma função de agregação. Se você não quiser agregação, use a palavra-chave`COLLECT_LIST`.

Example Exemplo

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
```

```
            "valueColumn": "all_votes"
        }
    }
}
```

SCALE

Dimensiona ou normaliza o intervalo de dados em uma coluna numérica.

Parâmetros

- `sourceColumn`— O nome de uma coluna existente.
- `strategy`— A operação a ser aplicada aos valores da coluna:
  - `MIN_MAX`— Redimensiona os valores em um intervalo de [0,1].
  - `SCALE_BETWEEN`— Redimensiona os valores em um intervalo de dois valores especificados.
  - `MEAN_NORMALIZATION`— Redimensiona os dados para ter uma média (μ) de 0 e desvio padrão (σ) de 1 dentro de um intervalo de [-1, 1].
  - `Z_SCORE`— Dimensiona linearmente os valores dos dados para ter uma média (μ) de 0 e desvio padrão (σ) de 1. Ideal para lidar com valores discrepantes.
- `targetColumn`— O nome de uma coluna para conter os resultados.

Example Exemplo

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

TRANSPÕEM

Converte todas as linhas selecionadas em colunas e as colunas em linhas.

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

Parâmetros

- `pivotColumns`— Uma JSON-encoded string representando uma lista de colunas cujas linhas serão convertidas em nomes de colunas.
- `valueColumns`— Uma JSON-encoded string representando uma lista de uma ou mais colunas a serem convertidas em linhas.
- `aggregateFunction`— O nome de uma função de agregação. Se você não quiser agregação, use a palavra-chave`COLLECT_LIST`.
- `newColumn`— A coluna para manter as colunas transpostas como valores.

Example Exemplo

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
            "newColumn": "Student"
        }
    }

}
```

UNION

Combina as linhas de dois ou mais conjuntos de dados em um único resultado.

Parâmetros

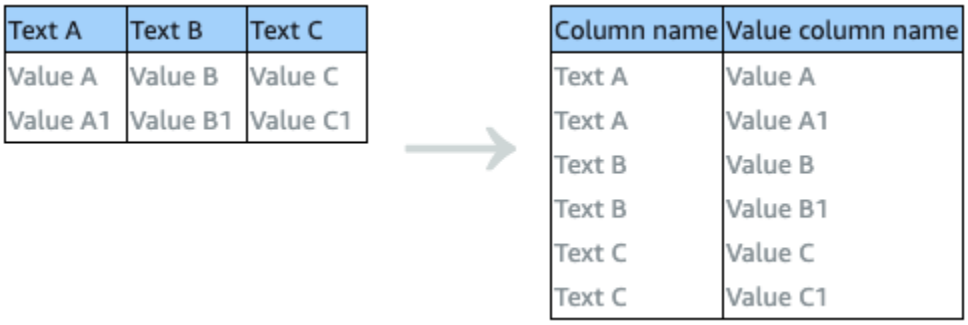
- `datasetsColumns`— Uma JSON-encoded string representando uma lista de todas as colunas nos conjuntos de dados.
- `secondaryDatasetNames`— Uma JSON-encoded string representando uma lista de um ou mais conjuntos de dados secundários.
- `secondaryInputs`— Uma JSON-encoded string representando uma lista de buckets do Amazon S3 e nomes de chaves de objetos que informam DataBrew onde encontrar o (s) conjunto (s) de dados secundário (s).
- `targetColumnNames`— Uma JSON-encoded string representando uma lista de nomes de colunas para os resultados.

Example Exemplo

```
{
    "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

Converte todos os valores da coluna em uma linha selecionada em linhas individuais com valores.



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

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de uma ou mais colunas a serem desarticuladas.
- `unpivotColumn`— A coluna de valor para a operação de despivô.
- `valueColumn`— A coluna para conter valores não dinâmicos.

Example Exemplo

```
{
    "Action": {
        "Operation": "UNPIVOT",
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
    }
}
```

Etapas da receita da ciência de dados

Use essas etapas da receita para tabular e resumir dados de diferentes perspectivas ou para realizar transformações avançadas.

Tópicos

BINARIZAÇÃO

Pega todos os valores em uma coluna de origem numérica selecionada, os compara com um valor limite e gera uma nova coluna com 1 ou 0 para cada linha.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `targetColumn`: o nome da nova coluna a ser criada.

  `threshold`— Número indicando o limite para atribuir o valor de 0 ou 1.

  `flip`— Opção de inverter a atribuição binária para que valores mais baixos sejam atribuídos 1 e valores mais altos sejam atribuídos 0. Quando o parâmetro flip é verdadeiro, valores menores ou iguais ao valor limite resultam em 1, e valores maiores que o valor limite resultam em 0.

Example Exemplo

```
{
    "Action": {
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
```

```
}
```

BUCKETIZAÇÃO

A bucketização (chamada de compartimento no console) pega os itens em uma coluna de valores numéricos, os agrupa em compartimentos definidos por intervalos numéricos e gera uma nova coluna que exibe o compartimento para cada linha. A bucketização pode ser feita usando parcelas ou porcentagens. O primeiro exemplo abaixo usa divisões e o segundo exemplo usa uma porcentagem.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `targetColumn`: o nome da nova coluna a ser criada.

  `bucketNames`— Lista de nomes de buckets.

  `splits`— Lista de níveis de balde. Os compartimentos são consecutivos, e um limite superior para um compartimento será um limite inferior para o próximo compartimento.

  `percentage`— Cada balde será descrito como uma porcentagem.

Example Exemplo de uso de divisões

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
        }
    }
}
```

Example Exemplo usando uma porcentagem

```
{
```

```
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

MAPEAMENTO_CATEGÓRICO

Mapeia um ou mais valores categóricos para valores numéricos ou outros

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `categoryMap`— Uma JSON-encoded string representando um mapa de valores para categorias.

  `deleteOtherRows`— Se`true`, todas as linhas não mapeadas serão removidas do conjunto de dados.

  `other`— Quando fornecidos, todos os valores não mapeados serão substituídos por esse valor.

  `keepOthers`— Se verdadeiro, todos os valores não mapeados permanecerão os mesmos.

  `mapType`— O tipo de dados da coluna mapeada.

  `targetColumn`— O nome de uma coluna para conter os resultados.

Example Exemplo

```
{
    "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

Cria n colunas numéricas, onde n é o número de valores exclusivos em uma variável categórica selecionada.

Por exemplo, considere uma coluna chamada`shirt_size`. As camisas estão disponíveis em tamanhos pequenos, médios, grandes ou extra grandes. Os dados da coluna podem ter a seguinte aparência.

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
M
```

Nesse cenário, há quatro valores distintos para`shirt_size`. Portanto, `ONE_HOT_ENCODING` gera quatro novas colunas. Cada nova coluna é nomeada`shirt_size_`*x*, onde *x* representa um `shirt_size` valor distinto.

Os resultados de `shirt_size` e as quatro colunas geradas têm a seguinte aparência.

```
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
```

A coluna que você especifica `ONE_HOT_ENCODING` pode ter no máximo dez (10) valores distintos.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente. A coluna pode ter no máximo 10 valores distintos.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SCALE

Dimensiona ou normaliza o intervalo de dados em uma coluna numérica.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `strategy`— A operação a ser aplicada aos valores da coluna:
  - `MIN_MAX`— Redimensiona os valores em um intervalo de [0,1]

- `SCALE_BETWEEN`— Redimensiona os valores em um intervalo de 2 valores especificados.
- `MEAN_NORMALIZATION`— Redimensiona os dados para ter uma média (μ) de 0 e desvio padrão (σ) de 1 dentro de um intervalo de [-1, 1]
- `Z_SCORE`— Escale linearmente os valores dos dados para ter uma média (μ) de 0 e desvio padrão (σ) de 1. Ideal para lidar com valores discrepantes.

- `targetColumn`— O nome de uma coluna para conter os resultados.

Example Exemplo

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

ASSIMETRIA

Aplica transformações em seus valores de dados para alterar a forma da distribuição e sua inclinação.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `targetColumn`: o nome da nova coluna a ser criada.

  `skewFunction`
  - `ROOT`— extrair a raiz do valor. A raiz pode ser fornecida no `value` parâmetro.

    `LOG`— valor base logarítmico. A base do log pode ser fornecida no `value` parâmetro.

    `SQUARE`— função quadrada

  `value`— Argumento da SkewFunction.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

TOKENIZAÇÃO

Divide o texto em unidades menores, ou símbolos, como palavras ou termos individuais.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `delimiter`— Um delimitador personalizado que aparece entre palavras tokenizadas. (O comportamento padrão é separar cada token por um espaço.)
- `expandContractions`— Se`ENABLED`, expande as palavras contraídas. Por exemplo: “não” se torna “não faça”.
- `stemmingMode`— Divide o texto em unidades ou símbolos menores, como palavras ou termos individuais em minúsculas. Dois modos de derivação estão disponíveis: `PORTER` |. `LANCASTER`
- `stopWordRemovalMode`— Remove palavras comuns como a, an, the e muito mais.
- `customStopWords`— Para`StopWordRemovalMode`, permite que você especifique uma lista personalizada de palavras irrelevantes.
- `targetColumn`— O nome de uma coluna para conter os resultados.

Example Exemplo

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

```
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

Funções matemáticas

A seguir, encontre tópicos de referência para funções matemáticas que funcionam com ações de receitas.

Tópicos

- ABSOLUTE
- ADD
- CEILING
- DEGREES
- DIVIDIR
- EXPOENTE
- FLOOR
- É_PAR
- É_ESTRANHO
- LN
- LOG
- MOD
- MULTIPLICAR
- NEGAR
- PI

- POWER
- RADIANS
- RANDOM
- ENTRE_ALEATÓRIO
- ROUND
- SIGN
- RAIZ_QUADRADA
- SUBTRAIR

ABSOLUTE

Retorna o valor absoluto do número de entrada em uma nova coluna. O valor absoluto é a distância entre o número e zero, independentemente de ser positivo ou negativo

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
        }
    }
}
```

ADD

Soma os valores da coluna de entrada em uma nova coluna, usando (`sourceColumn1+sourceColumn2`) ou (`sourceColumn1+value1`).

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `value1`— Um valor numérico.
- `sourceColumn2`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

CEILING

Retorna o menor número inteiro maior ou igual aos números decimais de entrada em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value1`— Um valor numérico.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "CEILING",
```

```
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

DEGREES

Converte radianos de um ângulo em graus e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

DIVIDIR

Divide um número de entrada por outro e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `value1`— Um valor numérico.
- `sourceColumn2`: o nome de uma coluna existente.
- `value2`— Um valor numérico.

- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
        }
    }
}
```

EXPOENTE

Retorna o número de Euler elevado ao enésimo grau em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_EXPONENT"
        }
    }
}
```

FLOOR

Retorna o maior número integral maior ou igual ao número de entrada em uma nova coluna.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `value`— Um valor numérico.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

É_PAR

Retorna um valor booleano em uma nova coluna que indica se a coluna ou o valor de origem é par. Se a coluna ou o valor de origem for decimal, o resultado será falso.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.
- `trueString`: uma string que indica se o valor é par.
- `falseString`— Uma string que indica se o valor não é par.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
        "Parameters": {
            "falseString": "Value is odd",
```

```
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

É_ESTRANHO

Retorna um valor booleano em uma nova coluna que indica se a coluna ou o valor de origem é ímpar. Se a coluna ou o valor de origem for decimal, o resultado será falso.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.
- `trueString`— Uma string que indica se o valor é ímpar.
- `falseString`— Uma string que indica se o valor não é ímpar.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
}
```

LN

Retorna o logaritmo natural (número de Euler) de um valor em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
        }
    }
}
```

LOG

Retorna o logaritmo de um valor em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.
- `base`— A base do logaritmo. O padrão é 10.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "LOG",
        "Parameters": {
            "base": "10",
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
    }
}
```

MOD

Retorna a porcentagem de um número de outro número em uma nova coluna.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

MULTIPLICAR

Multiplica dois números e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `value1`— Um valor numérico.
- `sourceColumn2`: o nome de uma coluna existente.
- `value2`— Um valor numérico.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

NEGAR

Nega um valor e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

Retorna o valor de pi (3,141592653589793) em uma nova coluna.

Parâmetros

- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

POWER

Retorna o valor de um número à potência do expoente em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`— Um número cujo valor deve ser aumentado.
- `targetColumn`: o nome da nova coluna a ser criada.
- `exponent`— A potência à qual o valor será elevado.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "POWER",
        "Parameters": {
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
```

```
    }
}
```

RADIANS

Converte graus em radianos (divide por 180/pi) e retorna o valor em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

RANDOM

Retorna um número aleatório entre 0 e 1 em uma nova coluna.

Parâmetros

- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "RANDOM",
```

```
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

ENTRE_ALEATÓRIO

Em uma nova coluna, retorna um número aleatório entre um limite inferior especificado (inclusive) e um limite superior especificado (inclusive).

Parâmetros

- `lowerBound`— O limite inferior do intervalo de números aleatórios.
- `upperBound`— O limite superior do intervalo de números aleatórios.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

ROUND

Arredonda um valor numérico para o número inteiro mais próximo em uma nova coluna. Ele arredonda para cima quando a fração é 0,5 ou mais.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

SIGN

Retorna uma nova coluna com -1 se o valor for menor que 0, 0 se o valor for 0 e +1 se o valor for maior que 0.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

RAIZ_QUADRADA

Retorna a raiz quadrada de um valor em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

SUBTRAIR

Subtrai um número do outro e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `value1`— Um valor numérico.
- `sourceColumn2`: o nome de uma coluna existente.
- `value2`— Um valor numérico.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "targetColumn": "weight_minus_10_kg",
```

```
            "value2": "10"
        }
    }
}
```

Funções agregadas

A seguir, encontre tópicos de referência para funções agregadas que funcionam com ações de receita.

Tópicos

- ANY
- AVERAGE
- CONTAGEM
- CONTAGEM_DISTINTA
- KTH_MAIOR
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN
- MODE
- DESVIO_PADRÃO
- SUM
- VARIANCE

ANY

Retorna todos os valores das colunas de origem selecionadas em uma nova coluna. Valores vazios e nulos são ignorados.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

AVERAGE

Calcula a média dos valores nas colunas de origem e retorna o resultado em uma nova coluna. Qualquer coisa que não seja numérica é ignorada.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

CONTAGEM

Retorna o número de valores das colunas de origem selecionadas em uma nova coluna. Valores vazios e nulos são ignorados.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

CONTAGEM_DISTINTA

Retorna o número total de valores distintos das colunas de origem selecionadas em uma nova coluna. Valores vazios e nulos são ignorados.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "COUNT_DISTINCT",
        "Parameters": {
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
    }
}
```

KTH_MAIOR

Retorna o késimo maior número das colunas de origem selecionadas em uma nova coluna.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.
- `value`— Um número representando k.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

Retorna o késimo maior número exclusivo das colunas de origem selecionadas em uma nova coluna.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

  `value`— Um número representando k.

Example Exemplo

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

```
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

Retorna o valor numérico máximo das colunas de origem selecionadas em uma nova coluna. Qualquer coisa que não seja numérica é ignorada.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
}
```

MEDIAN

Retorna a mediana, o número médio de um grupo ordenado de números, das colunas de origem selecionadas em uma nova coluna. Qualquer coisa que não seja numérica é ignorada.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

Retorna o valor mínimo das colunas de origem selecionadas em uma nova coluna. Qualquer coisa que não seja numérica é ignorada.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MIN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

Retorna o modo, o número que aparece com mais frequência, das colunas de origem selecionadas em uma nova coluna. Qualquer coisa que não seja numérica é ignorada. Para vários modos, o modo é calculado com a função modal.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

DESVIO_PADRÃO

Retorna o desvio padrão das colunas de origem selecionadas em uma nova coluna.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

SUM

Retorna a soma dos valores das colunas de origem selecionadas em uma nova coluna. Qualquer não-número é tratado como 0.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
    }
}
```

VARIANCE

Retorna a variação das colunas de origem selecionadas em uma nova coluna. A variância é definida como`Var(X) = [Sum ((X – mean(X))^2)]/Count(X)`.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando uma lista de colunas existentes.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

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

```
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

Funções de texto

A seguir, encontre tópicos de referência para funções de texto que funcionam com ações de receita.

Tópicos

- CHAR
- ENDS_WITH
- EXATO
- ACHAR
- LEFT
- LEN
- LOWER
- MESCLAR COLUNAS E VALORES
- APROPRIADO
- REMOVER_SÍMBOLOS
- REMOVE_WHITESPACE
- SEQÜÊNCIA DE CARACTERES DE REPETIÇÃO
- RIGHT
- LOCALIZAÇÃO_CERTA
- STARTS_WITH
- SEQÜÊNCIA_MAIOR_QUE
- SEQÜÊNCIA_MAIOR_QUE_IGUAL
- SEQÜÊNCIA_MENOS_QUE
- SEQÜÊNCIA_MENOS_QUE_IGUAL
- SUBSTRING
- TRIM
- UNICODE

- UPPER

CHAR

Retorna em uma nova coluna o caractere Unicode para cada inteiro na coluna de origem ou para um valor inteiro personalizado.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`— Um número inteiro que representa um valor Unicode.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
```

```
}
```

ENDS_WITH

`true`Retorna em uma nova coluna se um número especificado de caracteres mais à direita, ou string personalizada, corresponder a um padrão.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `pattern`— Uma expressão regular que deve corresponder ao final da string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

EXATO

Cria uma nova coluna preenchida com uma das seguintes opções:

- `True`se uma string em uma coluna (ou valor) corresponder exatamente a outra string em uma coluna (ou valor) diferente.

- `Falsese` não houver partida.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar somente uma das seguintes combinações:
>
> - Ambos`sourceColumn`*N*.
> - Um dos `sourceColumn`*N* e um dos`value`*N*.
> - Ambos`value`*N*.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "EXACT",
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

ACHAR

Pesquisando da esquerda para a direita, encontra cadeias de caracteres que correspondam a uma sequência de caracteres especificada da coluna de origem ou de um valor personalizado e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— Uma expressão regular para pesquisar.
- `position`— A posição do caractere para começar, a partir da extremidade esquerda da string.
- `ignoreCase`— Se`true`, ignore as diferenças de maiúsculas e minúsculas (entre maiúsculas e minúsculas) entre as letras. Para impor uma correspondência estrita, use `false` em vez disso.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

LEFT

Dado um número de caracteres, pega o número de caracteres mais à esquerda na string da coluna de origem ou da string personalizada e retorna o número especificado de caracteres mais à esquerda em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `position`— A posição do caractere para começar, a partir da extremidade esquerda da string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "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

Retorna em uma nova coluna o comprimento das strings da coluna de origem ou das strings personalizadas.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
        }
    }
}
```

LOWER

Converte todos os caracteres alfabéticos das cadeias de caracteres na coluna de origem ou das sequências personalizadas em minúsculas e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
    }
}
```

MESCLAR COLUNAS E VALORES

Concatena as cadeias de caracteres nas colunas de origem e retorna o resultado em uma nova coluna. Você pode inserir um delimitador entre os valores mesclados.

Parâmetros

- `sourceColumns`— Os nomes de duas ou mais colunas existentes, em JSON-encoded formato.
- `delimiter`: opcional. Um ou mais caracteres para colocar entre cada dois valores da coluna de origem.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
    }
}
```

APROPRIADO

Converte todos os caracteres alfabéticos das cadeias de caracteres na coluna de origem ou valores personalizados em maiúsculas e minúsculas e retorna o resultado em uma nova coluna.

No caso próprio, também chamado de maiúscula, a primeira letra de cada palavra é maiúscula e o restante da palavra é transformado em minúscula. Um exemplo é: The Quick Brown Fox pulou sobre a cerca

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "PROPER",
```

```
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_proper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "value": "MR. H. SMITH, ESQ.",
            "targetColumn": "formal_name_proper"
        }
    }
}
```

REMOVER_SÍMBOLOS

Remove caracteres que não são letras, números, caracteres latinos acentuados ou espaço em branco das cadeias de caracteres na coluna de origem ou das sequências personalizadas e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
```

```
    "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

Remove o espaço em branco das cadeias de caracteres na coluna de origem ou nas sequências personalizadas e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
```

```
    "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"
        }
    }
}
```

SEQÜÊNCIA DE CARACTERES DE REPETIÇÃO

Repete as cadeias de caracteres na coluna de origem ou no valor de entrada personalizado um número especificado de vezes e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `count`— O número de vezes para repetir a string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "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

Dado um número de caracteres, pega o número mais à direita das cadeias de caracteres da coluna de origem ou das cadeias personalizadas e retorna o número especificado de caracteres mais à direita em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `position`— A posição do caractere para começar, do lado direito da string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

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

LOCALIZAÇÃO_CERTA

Pesquisando da direita para a esquerda, encontra cadeias de caracteres que correspondam a uma sequência de caracteres especificada da coluna de origem ou de um valor personalizado e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `pattern`— Uma expressão regular para pesquisar.
- `position`— A posição do caractere para começar, a partir da extremidade direita da string.
- `ignoreCase`— Se`true`, ignore as diferenças de maiúsculas e minúsculas (entre maiúsculas e minúsculas) entre as letras. Para impor uma correspondência estrita, use `false` em vez disso.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

`true`Retorna em uma nova coluna se um número especificado de caracteres mais à esquerda, ou string personalizada, corresponder a um padrão.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `pattern`— Uma expressão regular que deve corresponder ao início da string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

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

```
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
        }
    }
}
```

SEQÜÊNCIA_MAIOR_QUE

Cria uma nova coluna preenchida com uma das seguintes opções:

- `True`se uma string em uma coluna (ou valor) for maior do que outra string em uma coluna (ou valor) diferente.
- `False`se não houver partida.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar somente uma das seguintes combinações:
>
> - Ambos`sourceColumn`*N*.
> - Um dos `sourceColumn`*N* e um dos`value`*N*.
> - Ambos`value`*N*.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
```

```
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
        }
    }
}
```

SEQÜÊNCIA_MAIOR_QUE_IGUAL

Cria uma nova coluna preenchida com uma das seguintes opções:

- `True`se uma string em uma coluna (ou valor) for maior ou igual a outra string em uma coluna (ou valor) diferente.
- `False`se não houver partida.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar somente uma das seguintes combinações:
>
> - Ambos`sourceColumn`*N*.
> - Um dos `sourceColumn`*N* e um dos`value`*N*.
> - Ambos`value`*N*.

Example Exemplo

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

```
        "Operation": "STRING_GREATER_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "nationality",
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

SEQÜÊNCIA_MENOS_QUE

Cria uma nova coluna preenchida com uma das seguintes opções:

- `True`se uma string em uma coluna (ou valor) for menor que outra string em uma coluna (ou valor) diferente.
- `False`se não houver partida.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar somente uma das seguintes combinações:
>
> - Ambos`sourceColumn`*N*.
> - Um dos `sourceColumn`*N* e um dos`value`*N*.
> - Ambos`value`*N*.

Example Exemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

SEQÜÊNCIA_MENOS_QUE_IGUAL

Cria uma nova coluna preenchida com uma das seguintes opções:

- `True`se uma string em uma coluna (ou valor) for menor ou igual a outra string em uma coluna (ou valor) diferente.
- `False`se não houver partida.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.
- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar somente uma das seguintes combinações:
>
> - Ambos`sourceColumn`*N*.
> - Um dos `sourceColumn`*N* e um dos`value`*N*.
> - Ambos`value`*N*.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

Retorna em uma nova coluna algumas ou todas as cadeias de caracteres especificadas na coluna de origem, com base nos valores de índice inicial e final definidos pelo usuário.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `startPosition`— A posição do caractere para começar, a partir da extremidade esquerda da string.
- `endPosition`— A posição do caractere com a qual terminar, a partir da extremidade esquerda da string.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
```

```
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
}
```

TRIM

Remove os espaços em branco à esquerda e à direita das cadeias de caracteres na coluna de origem ou nas cadeias personalizadas e retorna o resultado em uma nova coluna. Os espaços entre as palavras não são removidos.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "TRIM",
```

```
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
    }
}
```

UNICODE

Retorna em uma nova coluna o valor do índice Unicode para o primeiro caractere das cadeias de caracteres na coluna de origem ou para cadeias de caracteres personalizadas.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
```

```
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
        }
    }
}
```

UPPER

Converte todos os caracteres alfabéticos das cadeias de caracteres na coluna de origem ou das sequências personalizadas em maiúsculas e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

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

```
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
        }
    }
}
```

Perfis de data e hora

A seguir, encontre tópicos de referência para funções de data e hora que funcionam com ações de receita.

Tópicos

- CONVERT_TIMEZONE
- DATE
- DATE_ADD
- DATE_DIFF
- FORMATO_DATA
- DATE_TIME
- DAY
- HOUR
- MILLISECOND
- MINUTE
- MONTH
- NOME_DO_MÊS
- NOW
- QUARTO
- SECOND
- TIME
- HOJE
- HORÁRIO_UNIX
- FORMATO_TEMPO_UNIX
- DIA DA SEMANA
- NÚMERO_SEMANA

- YEAR

CONVERT_TIMEZONE

Converte um valor de tempo da coluna de origem em uma nova coluna com base em um fuso horário especificado.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente. A coluna de origem pode ser do tipo `stringdate`, `outimestamp`.
- `fromTimeZone`— Fuso horário do valor da fonte. Se nada for especificado, o fuso horário padrão será UTC.
- `toTimeZone`— Fuso horário a ser convertido para. Se nada for especificado, o fuso horário padrão será UTC.
- `targetColumn`— Um nome para a coluna recém-criada.
- `dateTimeFormat`: opcional. Uma string de formato para a data. Se o formato não for especificado, o formato padrão será usado:`yyyy-mm-dd HH:MM:SS`.

Example Exemplo

```
{
    "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

Cria uma nova coluna contendo o valor da data, a partir das colunas de origem ou dos valores fornecidos.

Parâmetros

- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se essa string não for especificada, o formato padrão seráyyyy-mm-dd HH:MM:SS.
- `dateTimeParameters`— Uma JSON-encoded string representando os componentes da data e hora:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Cada componente deve especificar um dos seguintes:
  - `sourceColumn`: o nome de uma coluna existente.
  - `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "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

Adiciona um ano, mês ou dia à data a partir de uma coluna ou valor de origem e cria uma nova coluna contendo os resultados.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `units`— Uma unidade de medida para ajustar a data. Os valores válidos são `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`,, `MINUTES` e.
- `dateAddValue`— O número de `units` a serem adicionados à data.
- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se não especificado, o formato padrão será `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
}
```

DATE_DIFF

Cria uma nova coluna contendo a diferença entre duas datas.

Parâmetros

- `sourceColumn1`: o nome de uma coluna existente.

- `sourceColumn2`: o nome de uma coluna existente.
- `value1`: uma sequência de caracteres para avaliar.
- `value2`: uma sequência de caracteres para avaliar.
- `units`— Uma unidade de medida para descrever a diferença entre as datas. Os valores válidos são `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`,, `MINUTES` e.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você só pode especificar uma das seguintes combinações:
>
> - Tanto de `sourceColumn1` quanto`sourceColumn2`.
> - Um de `sourceColumn1` ou `sourceColumn2` e um de `value1` ou`value2`.
> - Tanto de `value1` quanto`value2`.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
        }
    }
}
```

FORMATO_DATA

Cria uma nova coluna contendo uma data, em um formato específico, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`— Uma string para avaliar.
- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se não especificado, o formato padrão será `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplos

```
{
    "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

Cria uma nova coluna contendo o valor de data e hora, a partir das colunas de origem ou dos valores fornecidos.

Parâmetros

- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se essa string não for especificada, o formato padrão seráyyyy-mm-dd HH:MM:SS.
- `dateTimeParameters`— Uma JSON-encoded string representando os componentes da data e hora:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Cada componente deve especificar um dos seguintes:
  - `sourceColumn`: o nome de uma coluna existente.
  - `value`: uma sequência de caracteres para avaliar.

Example Exemplo

```
{
    "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

Cria uma nova coluna contendo o dia do mês, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

HOUR

Cria uma nova coluna contendo o valor da hora, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
}
```

MILLISECOND

Cria uma nova coluna contendo o valor de milissegundos de uma coluna de origem ou valor de entrada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente. A coluna de origem pode ser do tipo `stringdate`, `outimestamp`.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`— Um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
}
```

MINUTE

Cria uma nova coluna contendo o valor do minuto, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
}
```

MONTH

Cria uma nova coluna contendo o número do mês, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

NOME_DO_MÊS

Cria uma nova coluna contendo o nome do mês, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

NOW

Cria uma nova coluna contendo a data e a hora atuais no formatoyyyy-mm-dd HH:MM:SS.

Parâmetros

- `timeZone`— O nome de um fuso horário. Se nenhum fuso horário for especificado, o padrão será Tempo Coordenado Universal (UTC).
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
}
```

QUARTO

Cria uma nova coluna contendo o trimestre baseado em data a partir de uma string que representa uma data.

> **Note**
>
> Os trimestres são designados na nova coluna como 1, 2, 3 ou 4.
>
> - 1 é janeiro, fevereiro e março.
> - 2 é abril, maio e junho.
> - 3 é julho, agosto e setembro.
> - 4 é outubro, novembro e dezembro.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente. A coluna de origem pode ser do tipo `stringdate`, `outimestamp`.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`— Um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
}
```

SECOND

Cria uma nova coluna contendo o segundo valor, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

Cria uma nova coluna contendo o valor do tempo, a partir das colunas de origem ou dos valores fornecidos.

Parâmetros

- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se essa string não for especificada, o formato padrão seráyyyy-mm-dd HH:MM:SS.
- `dateTimeParameters`— Uma JSON-encoded string representando os componentes da data e hora:
  - `year`
  - `value`
  - `month`

- `day`
- `hour`
- `second`

Cada componente deve especificar um dos seguintes:

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.

- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

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

HOJE

Cria uma nova coluna contendo a data atual no formatoyyyy-mm-dd.

Parâmetros

- `timeZone`— O nome de um fuso horário. Se nenhum fuso horário for especificado, o padrão será Tempo Coordenado Universal (UTC).
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
        }
    }
}
```

HORÁRIO_UNIX

Cria uma nova coluna contendo um número representando a hora da época (horário Unix) — o número de segundos desde 1º de janeiro de 1970 — com base em uma coluna de origem ou valor de entrada. Se o fuso horário puder ser inferido, a saída estará nesse fuso horário. Caso contrário, a saída estará no Tempo Coordenado Universal (UTC).

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
        }
    }
}
```

FORMATO_TEMPO_UNIX

Converte a hora Unix de uma coluna de origem ou valor de entrada em um formato de data numérica especificado e retorna o resultado em uma nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`— Um número inteiro que representa um carimbo de data/hora da época do Unix.
- `dateTimeFormat`: opcional. Uma string de formato para a data, como ela deve aparecer na nova coluna. Se não especificado, o formato padrão será `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

DIA DA SEMANA

Cria uma nova coluna contendo o dia da semana, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

NÚMERO_SEMANA

Cria uma nova coluna contendo o número da semana (de 1 a 52), a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

YEAR

Cria uma nova coluna contendo o ano, a partir de uma string que representa uma data.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: um nome para a coluna recém-criada.

> **Note**
>
> Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
}
```

Funções de janela

A seguir, encontre tópicos de referência para funções de janela que funcionam com ações de receita.

Tópicos

FILL

Retorna uma nova coluna com base em uma coluna de origem especificada. Para quaisquer valores ausentes ou nulos na coluna de origem, `FILL` escolhe o valor não vazio mais recente em uma janela de linhas antes e depois do valor de origem em questão. O valor escolhido é então colocado na nova coluna.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
}
```

NEXT

Retorna uma nova coluna, em que cada valor representa um valor que está n linhas depois na coluna de origem.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- numRows— Um valor que representa n linhas anteriores na coluna de origem. Por exemplo, se numRows for 3, então NEXT usa o terceiro próximo `sourceColumn` valor como o novo `targetColumn` valor.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
}
```

ANTERIOR

Retorna uma nova coluna, em que cada valor representa um valor que está nas n linhas anteriores na coluna de origem.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRows`— Um valor que representa n linhas anteriores na coluna de origem. Por exemplo, se `numRows` for 3, então `PREV` usa o terceiro `sourceColumn` valor anterior como o novo `targetColumn` valor.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

MÉDIA_CONTÍNUA

Retorna em uma nova coluna a média contínua dos valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
    }
}
```

CONTABILIDADE_CONTAGEM_A

Retorna em uma nova coluna a contagem contínua de valores não nulos de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

Retorna em uma nova coluna o maior valor contínuo de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `value`— O valor para k.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
  "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

Retorna em uma nova coluna o valor único contínuo k th maior de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.

- `value`— O valor para k.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
  "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

Retorna em uma nova coluna o máximo contínuo de valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  numRowsBefore— Várias linhas antes da linha de origem atual, representando o início da janela.
- numRowsAfter— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_MAX",
        "Parameters": {
```

```
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

ROLLING_MIN

Retorna em uma nova coluna o mínimo contínuo de valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

MODO ROLANTE

Retorna em uma nova coluna o modo de rolagem (valor mais comum) de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- modeType — A função modal a ser aplicada à janela. Os valores válidos são `NONE`, `MINIMUM`, `MAXIMUM` e `AVERAGE`.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

ROLLING_STANDARD_DEVIATION

Retorna em uma nova coluna o desvio padrão contínuo dos valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

SOMA_CONTÍNUA

Retorna em uma nova coluna a soma contínua dos valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.

  `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

ROLLING_VARIANCE

Retorna em uma nova coluna a variação contínua dos valores de um número especificado de linhas antes para um número especificado de linhas após a linha atual na coluna especificada.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `numRowsBefore`— Várias linhas antes da linha de origem atual, representando o início da janela.
- `numRowsAfter`— Várias linhas após a linha de origem atual, representando o final da janela.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

Retorna em uma nova coluna um identificador de sessão baseado em uma janela criada pelos nomes das colunas das instruções “agrupar por” e “ordenar por”.

Parâmetros

- `groupByColumns`— Uma JSON-encoded string descrevendo as colunas “agrupar por”.
- `orderByColumns`— Uma JSON-encoded string descrevendo as colunas “ordenadas por”.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESSION

Retorna em uma nova coluna um identificador de sessão baseado em uma janela criada pelos nomes das colunas das instruções “agrupar por” e “ordenar por”.

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `units`— Uma unidade de medida para descrever a duração da sessão. Os valores válidos são `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`,, `MINUTES` e.
- `value`— O número de `units` para definir o período de tempo.
- `groupByColumns`— Uma JSON-encoded string descrevendo as colunas “agrupar por”.
- `orderByColumns`— Uma JSON-encoded string descrevendo as colunas “ordenadas por”.
- `targetColumn`: um nome para a coluna recém-criada.

Example Exemplo

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
```

```
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
}
```

Funções da Web

A seguir, encontre tópicos de referência para funções da web que funcionam com ações de receita.

Tópicos

- IP_TO_INT
- INT_PARA_IP
- PARÂMETROS_URL

IP_TO_INT

Converte o valor do Protocolo de Internet versão 4 (IPv4) da coluna de origem ou outro valor no valor inteiro correspondente na coluna de destino e retorna o resultado em uma nova coluna. Essa função funciona somente para IPv4.

Por exemplo, considere o seguinte endereço IP.

```
192.168.1.1
```

Se você usar esse valor como entrada para`IP_TO_INT`, o valor de saída será o seguinte.

```
3232235777
```

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_PARA_IP

Converte o valor inteiro da coluna de origem ou outro valor no valor IPv4 correspondente na coluna de destino e retorna o resultado em uma nova coluna. Essa função funciona somente para IPv4.

Por exemplo, considere o número inteiro a seguir.

```
167772410
```

Se você usar esse valor como entrada para`INT_TO_IP`, o valor de saída será o seguinte.

```
10.0.0.250
```

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

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

```
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
  }
]
```

PARÂMETROS_URL

Extrai parâmetros de consulta de uma string de URL, formata-os como um objeto JSON e retorna o resultado em uma nova coluna.

Por exemplo, considere o seguinte URL.

```
https://example.com/?firstParam=answer&secondParam=42
```

Se você usar esse valor como entrada paraURL_PARAMS, o valor de saída será o seguinte.

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parâmetros

- `sourceColumn`: o nome de uma coluna existente.
- `value`: uma sequência de caracteres para avaliar.
- `targetColumn`: o nome da nova coluna a ser criada.

Você pode especificar `sourceColumn` ou `value`, mas não os dois.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
```

```
    }
}
```

Outras funções

A seguir, encontre tópicos de referência para outras funções que funcionam com ações de receitas.

Tópicos

- AGLUTINAR
- GET_ACTION_RESULT
- GET_STEP_DATAFRAME

AGLUTINAR

Retorna em uma nova coluna o primeiro valor não nulo encontrado na matriz de colunas. A ordem das colunas listadas na função determina a ordem na qual elas são pesquisadas.

Parâmetros

- `sourceColumns`— Uma JSON-encoded string representando a lista de colunas existentes.
- `targetColumn`: o nome da nova coluna a ser criada.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

Busca o resultado de uma ação enviada anteriormente. Somente para uso na experiência interativa.

Parâmetros

- `actionId`— O ActionId retornado na SendProjectSessionAction resposta original.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "GET_ACTION_RESULT",
        "Parameters": {
            "actionId": "7",
        }
    }
}
```

GET_STEP_DATAFRAME

Busca o quadro de dados de uma etapa na receita do projeto. Somente para uso na experiência interativa. Usado com o ViewFrame parâmetro para paginar em um grande quadro de dados.

Parâmetros

- `stepIndex`— O índice da etapa na receita do projeto para a qual buscar o quadro de dados.

Example Exemplo

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

Cotas para AWS Glue DataBrew

Você pode ver suas cotas DataBrew de serviço no console AWS Service Quotas. Você também pode solicitar um aumento de cota, para qualquer cota ajustável.

Histórico do documento para AWS Glue DataBrew Guia do desenvolvedor

Versão atual da API: databrew-2017-07-25

A tabela a seguir descreve a documentação desta versão do AWS Glue DataBrew. Se quiser ser notificado quando o Guia do AWS Glue DataBrew desenvolvedor for atualizado, você pode assinar o feed RSS.

Alteração	Descrição	Data
`glue:GetCustomEntityType` adicionado às políticas AWS gerenciadas	Essa permissão é necessária para executar trabalhos AWS Glue DataBrew de perfil com PII-identification ativado. Para obter mais informações, consulte AWS Glue DataBrew atualizações nas políticas AWS gerenciadas.	20 de março de 2024
Support para vários algoritmos de hashing na transformação CRYPTOGRAPHIC_HASH	Agora você pode especificar um algoritmo de hash ao fazer o hash de valores em uma coluna. Para obter mais informações, consulte CRYPTOGRAPHIC_HASH.	11 de agosto de 2023
`glue:BatchGetCustomEntityTypes` adicionado às políticas AWS gerenciadas	Essa permissão é necessária para executar trabalhos AWS Glue DataBrew de perfil com PII-identification ativado. Para obter mais informações, consulte AWS Glue DataBrew atualizações nas políticas AWS gerenciadas.	9 de maio de 2022

Support para o formato de arquivo Apache ORC	DataBrew agora suporta o Apache ORC como formato de arquivo para fontes e saídas de DataBrew dados. Para obter mais informaçõ es, consulte Tipos de arquivo compatíveis com fontes de dados.	31 de março de 2022
Support para acesso entre contas ao AWS Glue Data Catalog Amazon S3	Agora você pode acessar tabelas do AWS Glue Data Catalog S3 de outras,Co ntas da AWS se uma política de recursos apropriada for criada no AWS Glue console. Depois de criar uma política, as tabelas relevantes do Catálogo de Dados S3 podem ser selecionadas como fontes de entrada ao criar um DataBrew conjunto de dados. Para obter mais informações, consulte Conexões suportadas para fontes e saídas de dados.	11 de março de 2022

Support para integração de console nativo com a Amazon AppFlow	DataBrew agora tem integração nativa do console com a Amazon AppFlow. Essa integração significa que você pode se conectar aos dados do Salesforce, Zendesk, Slack e outros aplicativos de software como serviço (SaaS). ServiceNow Você também pode se conectar a dados Serviços da AWS como Amazon S3 e Amazon Redshift. Para obter mais informações, consulte Conexões suportadas para fontes e saídas de dados.	18 de novembro de 2021
Support para regras de qualidade de dados	DataBrew agora oferece suporte à criação de regras de qualidade de dados, que são verificações de validação personalizáveis que definem os requisitos comerciais para dados específicos. Para obter mais informações, consulte Validando a qualidade dos dados em AWS Glue DataBrew.	18 de novembro de 2021

Support para instruções SQL personalizadas	DataBrew agora oferece suporte a instruções SQL personalizadas para recuperar dados do Amazon Redshift e do Snowflake. Esse suporte significa que você pode usar uma consulta específica para selecionar e limitar os dados retornados de tabelas grandes. Para obter mais informações, consulte Conexões suportadas para fontes e saídas de dados.	18 de novembro de 2021
Support para detecção de PII	DataBrew agora suporta a detecção de informações de identificação pessoal (PII). Isso oferece a opção de mascarar PII durante a preparação dos dados. Para obter mais informações, consulte Identificação e tratamento de informações de identificação pessoal (PII).	18 de novembro de 2021
Support para AWS regiões adicionais	DataBrew agora oferece suporte a AWS regiões adicionais. Para ver uma lista das regiões compatíveis, consulte AWS Glue DataBrew endpoints e cotas.	5 de outubro de 2021

Support para gravação de dados em tabelas do Lake Formation-based Amazon S3	DataBrew agora suporta a gravação de dados em tabelas AWS Glue Data Catalog do S3 com base em.AWS Lake Formation DataBrew agora também oferece suporte à gravação de dados no formato Tableau Hyper. Para obter mais informações, consulte Criação e trabalho com trabalhos de AWS Glue DataBrew receitas.	13 de agosto de 2021
Support para gravação de dados em destinos JDBC	DataBrew agora suporta a gravação de dados diretamente em JDBC-supported bancos de dados e armazéns de dados. Isso inclui Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database e PostgreSQL. Para obter mais informações, consulte Criação e trabalho com trabalhos de AWS Glue DataBrew receitas.	23 de julho de 2021
Support para especificar quais estatísticas de qualidade de dados são geradas para um trabalho de perfil	DataBrew agora suporta a especificação de quais estatísticas de qualidade de dados são geradas automaticamente para conjuntos de dados em um trabalho de perfil. Para obter mais informações, consulte Criação e trabalho com trabalhos de AWS Glue DataBrew receitas.	23 de julho de 2021

Support para gravação de conjuntos de dados no AWS Glue Data Catalog	DataBrew agora inclui suporte para gravar conjuntos de dados diretamente no AWS Glue Data Catalog. Você pode optar por armazenar conjuntos de dados criados a partir de trabalhos que executam suas receitas de preparação de dados nas tabelas Amazon S3, Amazon Redshift e Amazon RDS no Catálogo de Dados. As tabelas do RDS suportadas incluem aquelas para Amazon Aurora, RDS para Oracle, RDS para Microsoft SQL Server, RDS para MySQL e RDS para PostgreSQL.	30 de junho de 2021
Support para identificação de tipos de dados avançados	DataBrew agora inclui suporte para identificar e marcar automaticamente tipos de dados avançados para colunas, o que facilita a normalização de colunas que contêm determinados tipos de dados. Esses tipos de dados incluem número do Seguro Social, endereço de e-mail, número de telefone, sexo, cartão de crédito, URL, endereço IP, data e hora, moeda, CEP, país, região, estado e cidade.	30 de junho de 2021

Support para usar AppFlow a Amazon para transferir dados de aplicativos SaaS	DataBrew agora suporta o uso da Amazon AppFlow para transferir dados para o Amazon S3 a partir de aplicativos de software como serviço (SaaS) de terceiros , como Salesforce, Zendesk, Slack e. ServiceNow Para obter mais informações, consulte Conexões suportadas para fontes e saídas de dados.	29 de abril de 2021
Support para criar DataBrew conjuntos de dados com entrada de bancos de dados JDBC	DataBrew agora oferece suporte à criação de conjuntos de dados a partir de dados em JDBC-supported bancos de dados e armazéns de dados, incluindo Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database e PostgreSQL. Para obter mais informações, consulte Conexões suportadas para fontes e saídas de dados.	2 de abril de 2021
Support para mais Regiões da AWS	DataBrew agora oferece suporte adicional Regiões da AWS. Para ver uma lista das regiões compatíveis, consulte AWS Glue DataBrew endpoints e cotas.	28 de janeiro de 2021

Novas transformações para lidar com a duplicação	Quatro novas transformações para lidar com a duplicação foram adicionadas ao DataBrew console e à API. Para obter mais informações, consulte DELETE_DUPLICATE_ROWS, FLAG_DUPLICATE_ROWS, FLAG_DUPLICATES_IN_COLUMNe REMOVE_DUPLICATES nas etapas da receita de qualidade de dados.	28 de janeiro de 2021
Delimitadores CSV adicionais	DataBrew agora oferece suporte a delimitadores adicionais além de vírgulas em arquivos de valores separados por vírgula (CSV) usados para criar conjuntos de dados. DataBrew Para obter mais informações, consulte Criação e uso de AWS Glue DataBrew conjuntos de dados.	28 de janeiro de 2021
DataBrew extensão para JupyterLab	Agora você pode usar AWS Glue DataBrew como uma extensão em JupyterLab. Para obter mais informações, consulte Usando DataBrew como uma extensão em JupyterLab.	20 de novembro de 2020
Nova ferramenta de preparação de dados:AWS Glue DataBrew	Esta é a primeira versão do Guia do desenvolvedor do AWS Glue DataBrew.	11 de novembro de 2020

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.