aws

Unable to locate subtitle

AWS Glue DataBrew Guía para desarrolladores

AWS Glue DataBrew Guía para desarrolladores: ***Unable to locate subtitle***

Table of Contents

¿Qué es?AWS Glue DataBrew?

AWS Glue DataBrew es una herramienta visual de preparación de datos que permite a los usuarios limpiar y normalizar los datos sin necesidad de escribir código. Su uso DataBrew ayuda a reducir el tiempo que se tarda en preparar los datos para el análisis y el aprendizaje automático (ML) hasta en un 80 por ciento, en comparación con la preparación de datos desarrollada a medida. Puede elegir entre más de 250 transformaciones listas para usar para automatizar las tareas de preparación de datos, como filtrar anomalías, convertir los datos a formatos estándar y corregir valores no válidos.

Gracias a DataBrew ello, los analistas de negocios, los científicos de datos y los ingenieros de datos pueden colaborar más fácilmente para obtener información a partir de datos sin procesar. Como no DataBrew tiene servidores, independientemente de su nivel técnico, puede explorar y transformar terabytes de datos sin procesar sin necesidad de crear clústeres ni administrar ninguna infraestructura.

Con la DataBrew interfaz intuitiva, puede descubrir, visualizar, limpiar y transformar datos sin procesar de forma interactiva. DataBrew hace sugerencias inteligentes para ayudarlo a identificar los problemas de calidad de los datos que pueden ser difíciles de encontrar y cuya solución puede llevar mucho tiempo. DataBrew Al preparar los datos, puede utilizar su tiempo para actuar en función de los resultados e iterarlos más rápidamente. Puede guardar la transformación como pasos de una receta, que puede actualizar o reutilizar más adelante con otros conjuntos de datos e implementarla de forma continua.

La siguiente imagen muestra cómo DataBrew funciona a un alto nivel.

Para usarlo DataBrew, debe crear un proyecto y conectarse a sus datos. En el espacio de trabajo del proyecto, los datos se muestran en una interfaz visual similar a una cuadrícula. Aquí puede explorar los datos y ver las distribuciones de valores y los gráficos para comprender su perfil.

Para preparar los datos, puede elegir entre más de 250 transformaciones de apuntar y hacer clic. Estas incluyen la eliminación de valores nulos, la sustitución de los valores faltantes, la corrección de las incoherencias del esquema, la creación de columnas basadas en funciones y muchas más. También puedes usar las transformaciones para aplicar técnicas de procesamiento del lenguaje natural (PNL) para dividir oraciones en frases. Las vistas previas inmediatas muestran una parte de los datos antes y después de la transformación, por lo que puede modificar la receta antes de aplicarla a todo el conjunto de datos.

Una vez DataBrew ejecutada la receta en el conjunto de datos, el resultado se almacena en Amazon Simple Storage Service (Amazon S3). Una vez que el conjunto de datos limpio y preparado esté en Amazon S3, otro sistema de almacenamiento o administración de datos podrá ingerirlo.

Conceptos y términos fundamentales en AWS Glue DataBrew

A continuación, encontrará una descripción general de los conceptos y la terminología principales en AWS Glue DataBrew. Después de leer esta sección, consulte Introducción al AWS Glue DataBrew, que explica el proceso de creación de proyectos y conexión de conjuntos de datos y tareas en ejecución.

Temas

Proyecto

El espacio de trabajo interactivo de preparación de datos DataBrew se denomina proyecto. Mediante un proyecto de datos, se administra una colección de elementos relacionados: datos, transformaciones y procesos programados. Como parte de la creación de un proyecto, se elige o se crea un conjunto de datos en el que trabajar. A continuación, creas una receta, que es un conjunto de instrucciones o pasos sobre los que DataBrew quieres actuar. Estas acciones transforman los datos sin procesar en un formulario que está listo para ser consumido por la canalización de datos.

Conjunto de datos

Por conjunto de datos se entiende simplemente un conjunto de datos: filas o registros que se dividen en columnas o campos. Cuando creas un DataBrew proyecto, te conectas a los datos que deseas transformar o preparar, o los cargas. DataBrew puede trabajar con datos de cualquier fuente, importados de archivos formateados y se conecta directamente a una lista cada vez mayor de almacenes de datos.

DataBrewEn efecto, un conjunto de datos es una conexión de solo lectura a sus datos. DataBrew recopila un conjunto de metadatos descriptivos para hacer referencia a los datos. No se puede modificar ni almacenar ningún dato real DataBrew. Para simplificar, utilizamos conjunto de datos para referirnos tanto al conjunto de datos real como a los DataBrew usos de los metadatos.

Fórmula

En DataBrew, una receta es un conjunto de instrucciones o pasos para los datos sobre los que DataBrew quieres actuar. Una receta puede contener muchos pasos y cada paso puede contener muchas acciones. Utilice las herramientas de transformación de la barra de herramientas para configurar todos los cambios que desee realizar en los datos. Más adelante, cuando esté listo

para ver el producto final de la receta, asigne este trabajo DataBrew y lo programe. DataBrew almacena las instrucciones sobre la transformación de datos, pero no almacena ninguno de sus datos reales. Puede descargar recetas y reutilizarlas en otros proyectos. También puedes publicar varias versiones de una receta.

Trabajo

DataBrew se encarga de transformar sus datos ejecutando las instrucciones que configuró al preparar una receta. El proceso de ejecutar estas instrucciones se denomina trabajo. Un trabajo puede poner en práctica sus recetas de datos de acuerdo con un cronograma preestablecido. Sin embargo, no está limitado a un horario. También puede ejecutar trabajos a pedido. Si quieres perfilar algunos datos, no necesitas una receta. En ese caso, solo tiene que configurar un trabajo de perfil para crear un perfil de datos.

Linaje de datos

DataBrew rastrea los datos en una interfaz visual para determinar su origen, lo que se denomina linaje de datos. Esta vista muestra cómo fluyen los datos a través de diferentes entidades de donde provienen originalmente. Puede ver su origen, otras entidades que le influyeron, qué pasó con ellos a lo largo del tiempo y dónde se almacenaron.

Perfil de datos

Al perfilar sus datos, DataBrew crea un informe denominado perfil de datos. Este resumen proporciona información sobre la forma actual de los datos, incluido el contexto del contenido, la estructura de los datos y sus relaciones. Puede crear un perfil de datos para cualquier conjunto de datos ejecutando un trabajo de perfil de datos.

Integraciones de productos y servicios

Utilice esta sección para saber con qué productos y servicios se integran DataBrew.

DataBrew funciona con los siguientes AWS servicios de redes, administración y gobierno:

- Amazon CloudFront
- AWS CloudFormation
- AWS CloudTrail

- Amazon CloudWatch
- AWS Step Functions

DataBrew funciona con los siguientes lagos AWS de datos y almacenes de datos:

- AWS Lake Formation
- Amazon S3

DataBrew admite los siguientes formatos y extensiones de archivo para cargar datos.

Formato	Extensión de archivo (opcional)	Extensiones para archivos comprimidos (obligatorias)

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

Libro de trabajo de Microsoft Excel	`.xlsx`	Sin soporte de compresión

JSON (documento JSON y líneas JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

Formato	Extensión de archivo (opcional)	Extensiones para archivos comprimidos (obligatorias)

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

DataBrew escribe los archivos de salida en Amazon S3 y admite los siguientes formatos y extensiones de archivo.

Formato	Extensión de archivo (sin comprimir)	Extensiones de archivo (comprimidas)
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	No compatible	`.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	Extensión de archivo (sin comprimir)	Extensiones de archivo (comprimidas)
XML	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`
JSON (solo en formato de líneas JSON)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	No compatible	No aplicable

Configuración AWS Glue DataBrew

Antes de empezar AWS Glue DataBrew, debe configurar algunos permisos, un usuario y un rol. Comience por realizar los siguientes pasos:

1. Crear una AWS cuenta según sea necesario y crear políticas AWS Identity and Access Management(de IAM) que permitan a los usuarios ejecutar DataBrew:
   - Crear una AWS cuenta nueva y añadir un usuario. Para obtener más información, consulte Configurar un nuevo AWS inscrita.
   - Añadir una política de IAM para un usuario de consola. Un usuario con estos permisos puede acceder DataBrew en Consola de administración de AWS.
   - Añadir permisos para los recursos de datos de un rol de IAM. Un rol de IAM con estos permisos puede acceder a los datos en nombre del usuario.

   Debe ser administrador de IAM para crear usuarios, funciones y políticas.
2. Añadir usuarios o grupos para DataBrew. Un usuario o grupo con los permisos correctos adjuntos puede acceder DataBrew a la consola.
3. Añadir un rol con permisos para acceder a los datos DataBrew. Un rol con los permisos correctos puede acceder a los datos en nombre del usuario.

Configurar un nuevo AWS inscrita

Si no tienes una AWS cuenta, regístrate y crea un AWS usuario administrador de IAM.

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearla.

Para suscribirse a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como

práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar Tareas que requieren acceso de usuario raíz.

Para crear un usuario administrador, elija una de las siguientes opciones.

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
En IAM Identity Center (recomendado)	Usar credenciales a corto plazo para acceder a AWS. Esto se ajusta a las prácticas recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulta Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.	Siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center.	Configure el acceso programático configurando el AWS CLI que se utilizará AWS IAM Identity Center en la Guía del AWS Command Line Interface usuario.
En IAM (no recomendado)	Usar credenciales a largo plazo para acceder a AWS.	Siguiendo las instrucciones de Crear un usuario de IAM para acceso de emergencia de la Guía del usuario de IAM.	Configure el acceso programático mediante Administrar las claves de acceso de los usuarios de IAM en la Guía del usuario de IAM.

Para obtener más información, consulte los siguientes temas de la guía del usuario de IAM:

- ¿Qué es IAM?
- Cómo configurarse con IAM
- Crear un usuario y un grupo de administración (consola)

Configuración del AWS CLI

Si planea usar JupyterLab o la DataBrew API, asegúrese de instalar AWS Command Line Interface(AWS CLI). No la necesitas para usar la DataBrew consola ni para realizar los pasos de los ejercicios de introducción.

Para configurar el AWS CLI

1. Descargue y configure el AWS CLI siguiendo los pasos que se indican a continuación:
   - Instalación de la AWS CLI
   - Conceptos básicos de configuración
2. Compruebe la configuración introduciendo el siguiente DataBrew comando en la línea de comandos.

   ```
   aws databrew help
   ```

   Si esta instrucción devuelve el error "`aws: error: argument command: Invalid choice`" seguido de una larga lista de servicios, desinstálelos AWS CLI y vuelva a instalarlos. Esta acción no sobrescribe la configuración existente.

AWS CLI los comandos utilizan la AWS región predeterminada de la configuración, a menos que la defina con un parámetro o un perfil. Puede añadir el `--region` parámetro a cada comando.

Si lo prefiere, puede añadir un perfil con nombre en `~/.aws/config` o `%UserProfile%/.aws/config` (en Microsoft Windows). Los perfiles con nombre asignado también pueden conservar otros ajustes, como se muestra en el siguiente ejemplo.

```
[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
```

Configuración AWS Identity and Access Management Permisos (IAM)

Antes de empezar, debe configurar algunas cosas en IAM. Debe ser administrador o contar con la ayuda de uno de ellos. Sin embargo, si tiene una cuenta con acceso de administrador, puede realizar estas tareas usted mismo. En esta sección encontrará instrucciones sencillas para cada tarea.

A continuación se presenta un resumen de lo que debe hacer:

- Como parte de este proceso, añada un usuario. No tiene que añadir un usuario nuevo, puede utilizar uno existente. Adjunta DataBrew permisos para que el usuario pueda abrir la DataBrew consola.
- Crear un rol de IAM. Un rol permite ciertas acciones y otorga permisos cuando se usa, dentro de ciertos límites. Por ejemplo, solo funciona para los usuarios de tu AWS cuenta. Puedes añadir más limitaciones más adelante.
- Cree la política o las políticas de IAM que necesite. Una política es una lista de cosas que un usuario puede hacer. Para crear una política, abra otra página de la consola y pegue el texto de un archivo que descargue.

> **Note**
>
> Lo que ofrecemos aquí es información básica de configuración. Le recomendamos que dedique un tiempo a personalizar sus permisos para que se adapten a sus necesidades de seguridad y conformidad. Si necesita ayuda, póngase en contacto con su administrador o con AWS Support.

Para añadir los permisos necesarios

1. Cree políticas de IAM que permitan a los usuarios ejecutar DataBrew de la siguiente manera:

   - Añada una política de IAM personalizada para un usuario de consola. Si no necesitas una política personalizada, puedes elegir la política AWS gestionada en su lugar. Solo tienes que añadirla al usuario en el paso 2. Un usuario con estos permisos puede acceder a la consola DataBrew de servicio.

- Agregue permisos para los recursos de datos. Un rol de IAM con estos permisos puede acceder a los datos en nombre del usuario.

Debe ser administrador para crear usuarios, roles y políticas.

2. Agregue usuarios o grupos para DataBrew. Un usuario o grupo con los permisos correctos adjuntos puede acceder a la DataBrew consola.
3. Agregue un rol con permisos para acceder a los datos DataBrew. Un rol con los permisos correctos puede acceder a los datos en nombre del usuario.

Configurar las políticas de IAM para DataBrew

Las políticas de IAM se utilizan para gestionar los permisos. Una política facilita la adición de todos los permisos relacionados de una sola vez, en lugar de hacerlo de uno en uno.

Le recomendamos que cree las políticas con los mismos nombres que le proporcionamos. Usamos los nombres que se muestran a continuación para estas políticas en toda la documentación. El uso de estos nombres también facilita el contacto con Support si alguna vez necesita ponerse en contacto con AWS Support. Sin embargo, puede optar por cambiar tanto los nombres de las políticas como su contenido. Para obtener más información sobre las políticas de IAM, consulte Crear una política gestionada por el cliente en la Guía del usuario de IAM.

Después de crear las políticas necesarias para su uso DataBrew, debe adjuntarlas a los usuarios y roles. Más adelante en esta sección se explica cómo hacerlo.

Temas

- Añadir una política de IAM para un usuario de consola
- Añadir permisos para los recursos de datos de un rol de IAM
- Configuración de las políticas de IAM para DataBrew

Añadir una política de IAM para un usuario de consola

La configuración de los permisos de un usuario para el Consola de administración de AWS es opcional, pero si necesita acceso a la consola, realice primero este paso.

Para configurar los permisos de acceso a DataBrew la consola, elige una de las siguientes opciones:

- Usa la política gestionada por AWS:`AwsGlueDataBrewFullAccessPolicy`. Si eliges esta opción, pasa a la siguiente política,Añadir permisos para los recursos de datos de un rol de IAM.
- Cree la política descrita en esta sección,`AwsGlueDataBrewCustomUserPolicy`. Esta opción le permite personalizar la política con requisitos de seguridad personalizados adicionales.

La siguiente política concede los permisos necesarios para ejecutar la DataBrew consola. Los permisos se proporcionan mediante IAM.

Para definir la política de AwsGlueDataBrewCustomUserPolicy IAM para DataBrew (consola)

1. Descarga el JSON para la política de `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. https://console.aws.amazon.com/iam/
3. En el panel de navegación, seleccione Políticas.
4. Para cada política, elija Crear política.
5. En la pantalla Crear política, vaya a la pestaña JSON.
6. Copia la declaración JSON de la política que descargaste. Pégala sobre la declaración de muestra en el editor.
7. Comprueba que la política se adapte a tu cuenta, a los requisitos de seguridad y a AWS los recursos necesarios. Si necesita realizar cambios, puede hacerlos en el editor.
8. Elija Revisar política.

Para definir la política AwsGlueDataBrewCustomUserPolicy de IAM para DataBrew (AWS CLI)

1. Descargue el JSON para la política de `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Personalice la política como se describe en el primer paso del procedimiento anterior.
3. Ejecute el siguiente comando para crear la política.

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

Añadir permisos para los recursos de datos de un rol de IAM

Para conectarse a los datos,AWS Glue DataBrew debe tener una función de IAM que pueda transferir en nombre del usuario. A continuación, encontrará información sobre cómo crear la política que luego asociará a una función de IAM.

La `AwsGlueDataBrewDataResourcePolicy` política otorga los permisos necesarios para conectarse a los datos utilizando DataBrew. Para cualquier operación que tenga acceso a datos de otro AWS recurso, como acceder a sus objetos en Amazon S3, DataBrew necesitará permiso para acceder al recurso en su nombre.

Para definir la política de AwsGlueDataBrewDataResourcePolicy IAM para DataBrew (consola)

1. Descargue el JSON para. `AwsGlueDataBrewDataResourcePolicy`
2. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en https://console.aws.amazon.com/iam/.
3. En el panel de navegación, seleccione Políticas.
4. Para cada política, elija Crear política.
5. En la pantalla Crear política, vaya a la pestaña JSON.
6. Copia la declaración JSON de la política que descargaste. Pégala sobre la declaración de muestra en el editor.
7. Comprueba que la política se adapte a tu cuenta, a los requisitos de seguridad y a AWS los recursos necesarios. Si necesita realizar cambios, puede hacerlos en el editor.
8. Elija Revisar política.

Para definir la política AwsGlueDataBrewDataResourcePolicy de IAM para DataBrew (AWS CLI)

1. Descargue el JSON para. `AwsGlueDataBrewDataResourcePolicy`
2. Personalice la política como se describe en el primer paso del procedimiento anterior.
3. Ejecute el siguiente comando para crear la política.

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

Configuración de las políticas de IAM para DataBrew

A continuación, encontrará detalles y ejemplos sobre las políticas de IAM que puede utilizar. DataBrew Los detalles sobre las políticas básicas se proporcionan aquí. Además, hay más ejemplos que no es obligatorio utilizar DataBrew. Son configuraciones adicionales que puede utilizar en determinadas situaciones.

Temas

AwsGlueDataBrewCustomUserPolicy

La `AwsGlueDataBrewCustomUserPolicy` política concede la mayoría de los permisos necesarios para usar la DataBrew consola. Algunos de los recursos que se especifican en esta política se refieren a los servicios que utilizan DataBrew. Estos incluyen los nombres de AWS Glue Data Catalog los buckets de Amazon S3, CloudWatch los registros de Amazon y AWS KMS los recursos. Es similar a la política AWS administrada denominada. `AwsGlueDataBrewFullAccessPolicy`

En la siguiente tabla se describen los permisos que esta política concede.

Action	Resource	Descripción
`"databrew:*"`	`"*"`	Otorga permiso para ejecutar todas las operacion es de DataBrew la API.

`"glue:GetDatabases"`
`"glue:GetPartitions"`
`"glue:GetTable"`
`"glue:GetTables"`
	`"*"`	Permite enumerar AWS Glue bases de datos y tablas.

Action	Resource	Descripción
"glue:GetDataCatalogEncryptionSettings"		
"dataexchange:ListDataSets" "dataexchange:ListDataSetRevisions" "dataexchange:ListRevisionAssets" "dataexchange:CreateJob" "dataexchange:StartJob" "dataexchange:GetJob"	"*"	Permite enumerar los recursos de AWS Data Exchange en conjuntos de datos.
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	Permite enumerar las AWS KMS claves que se utilizarán para cifrar los resultados del trabajo.
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	Permite cifrar el resultado del trabajo.
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionConfiguration"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Permite publicar buckets de Amazon S3 para proyectos, conjuntos de datos y trabajos. Permite enviar archivos de salida a S3.

Action	Resource	Descripción
`"sts:GetCallerIdentity"`	`"*"`	Obtenga información sobre la persona que llama actualmente.
`"cloudtrail:LookupEvents",`	`"*"`	Permita enumerar AWS CloudTrail eventos para conjuntos de datos (linaje de datos).
`"iam:ListRoles"` `"iam:GetRole"`	`"*"`	Permite enumerar las funciones de IAM para utilizarlas en proyectos y trabajos.

AwsGlueDataBrewDataResourcePolicy

La `AwsGlueDataBrewDataResourcePolicy` política concede los permisos necesarios para conectarse a los datos y realizar la configuración DataBrew.

En la siguiente tabla se describen los permisos que esta política concede.

Action	Resource	Descripción
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Le permite obtener una vista previa de sus archivos.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permite enviar los archivos de salida a S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*",`	Permite eliminar un objeto creado por DataBrew.

Action	Resource	Descripción
	`"arn:aws:s3:::bucket_name"`	
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite enumerar buckets de Amazon S3 de proyectos, conjuntos de datos y trabajos.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Permite descifrar conjuntos de datos cifrados.
`"kms:GenerateDataKey"`	`"arn:aws:kms:::key/key_ids"`	Permite cifrar la salida del trabajo.

`"ec2:DescribeVpcEndpoints"`
`"ec2:DescribeRouteTables"`
`"ec2:DeleteNetworkInterface"`
`"ec2:DescribeNetworkInterfaces"`
`"ec2:DescribeSecurityGroups"`
`"ec2:DescribeSubnets"`
`"ec2:DescribeVpcAttribute"`
`"ec2:CreateNetworkInterface"`
	`"*"`	Permite configurar elementos de red de Amazon EC2, como nubes privadas virtuales (VPC), al ejecutar trabajos y proyectos.

Action	Resource	Descripción
`"ec2:DeleteNetworkInterface"`	`"*"`	Permite eliminar una interfaz de red en una VPC.

`"ec2:CreateTags"`
 	
`"ec2:DeleteTags"`
	`"arn:aws:ec2:::network-interface/*", "arn:aws:ec2:::security-group/*"`	Permite crear y eliminar etiquetas.
		
		Necesitará estos permisos si utiliza un catálogo de AWS Glue datos con una VPC habilitada. DataBrew transfiere los datos AWS Glue para ejecutar sus trabajos y proyectos. Estos permisos permiten etiquetar los recursos de Amazon EC2 creados para los puntos finales de desarrollo.AWS Glue etiqueta las interfaces de red, los grupos de seguridad y las instancias de Amazon EC2 con. `aws-glue-service-resource`

`"logs:CreateLogGroup"`
 	
`"logs:CreateLogStream"`
 	
`"logs:PutLogEvents"`
	`"arn:aws:logs:::log-group:/aws-glue-databrew/*"`	Permite escribir registros en Amazon CloudWatch Logs
		
		DataBrew escribe registros en grupos de registros cuyos nombres comienzan por`aws-glue-databrew` .

Action	Resource	Descripción
`"lakeformation:GetDataAccess"`	`"*"`	Permite el acceso a AWS Lake Formation, siempre `"Glue":"GetTable"` que también esté permitido El uso de Lake Formation requiere una configuración adicional en la consola de Lake Formation.

Política de IAM para usar objetos de Amazon S3 con DataBrew

La `AwsGlueDataBrewSpecificS3BucketPolicy` política concede los permisos necesarios para acceder a S3 en nombre de los usuarios no administrativos.

Personalice la política de la siguiente manera:

1. Sustituya las rutas de Amazon S3 en la política para que apunten a las rutas que desee utilizar. En el texto de ejemplo, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* representa un objeto o archivo específico. *`BUCKET-NAME-2/`*representa todos los objetos (*) cuyo nombre de ruta comienza por`BUCKET-NAME-2/`. Actualícelos para asignar un nombre a los depósitos que está utilizando.
2. (Opcional) Utilice caracteres comodín en las rutas de Amazon S3 para restringir aún más los permisos. Para obtener más información, consulte Elementos de la política de IAM: Variables y etiquetas en la Guía del usuario de IAM.

   Práctica recomendada de seguridad: Para evitar el acceso no autorizado a buckets de Amazon S3 con nombres similares en otras AWS cuentas, incluya la clave de `aws:ResourceAccount` condición en su política. Esto garantiza que solo DataBrew pueda acceder a los depósitos de su propia AWS cuenta, incluso cuando utilice ARN de recursos comodín. Añada la siguiente condición a sus declaraciones de política:

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

`123456789012`Sustitúyala por tu ID AWS de cuenta real.

Como parte de esto, puedes restringir los permisos para las acciones `s3:PutObject` y`s3:PutBucketCORS`. Estas acciones solo son necesarias para los usuarios que crean DataBrew proyectos, ya que esos usuarios deben poder enviar los archivos de salida a S3.

Para obtener más información y ver algunos ejemplos de lo que puede añadir a una política de IAM para Amazon S3, consulte Ejemplos de políticas de buckets en la Guía para desarrolladores de Amazon S3.

En la siguiente tabla se describen los permisos que esta política concede.

Action	Resource	Descripción
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Le permite previsualizar sus archivos.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite enviar los archivos de salida a S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite borrar un objeto.

Para definir la política AwsGlueDataBrewSpecificS3BucketPolicy de IAM para DataBrew (consola)

1. Descarga el JSON para la política de `AwsGlueDataBrewSpecificS3BucketPolicy`IAM.
2. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. https://console.aws.amazon.com/iam/
3. En el panel de navegación, seleccione Políticas.

4. Para cada política, elija Crear política.
5. En la pantalla Crear política, vaya a la pestaña JSON.
6. Pegue la declaración JSON de la política sobre la declaración de ejemplo en el editor.
7. Comprueba que la política se adapte a tu cuenta, a los requisitos de seguridad y a AWS los recursos necesarios. Si necesita realizar cambios, puede hacerlos en el editor.
8. Elija Revisar política.

Para definir la política AwsGlueDataBrewSpecificS3BucketPolicy de IAM para DataBrew (AWS CLI)

1. Descargue el JSON para. `AwsGlueDataBrewSpecificS3BucketPolicy`
2. Personalice la política como se describe en el primer paso del procedimiento anterior.
3. Ejecute el siguiente comando para crear la política.

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

Política de IAM para utilizar el cifrado con DataBrew

La `AwsGlueDataBrewS3EncryptedPolicy` política concede los permisos necesarios para acceder a los objetos de S3 cifrados con AWS Key Management Service(AWS KMS) en nombre de usuarios no administrativos.

Personalice la política de la siguiente manera:

1. Sustituya las rutas de Amazon S3 en la política para que apunten a las rutas que desee utilizar. En el texto de ejemplo, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* representa un objeto o archivo específico. *`BUCKET-NAME-2/`*representa todos los objetos (*) cuyo nombre de ruta comienza por`BUCKET-NAME-2/`. Actualícelos para asignar un nombre a los depósitos que está utilizando.
2. (Opcional) Utilice caracteres comodín en las rutas de Amazon S3 para restringir aún más los permisos. Para obtener más información, consulte Elementos de la política de IAM: variables y etiquetas.

   Como parte de ello, puede restringir los permisos para las acciones `s3:PutObject` y`s3:PutBucketCORS`. Estas acciones solo son necesarias para los usuarios que crean DataBrew proyectos, ya que esos usuarios deben poder enviar los archivos de salida a S3.

Para obtener más información y ver algunos ejemplos de lo que puede añadir a una política de IAM para Amazon S3, consulte Ejemplos de políticas de buckets.

3. Busque los siguientes ARN de recursos en el `ToUseKms` archivo.

```
  "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. Cambie la AWS cuenta de ejemplo por su número de AWS cuenta (sin guiones).
5. Cambie la lista de ejemplos para incluir en su lugar las funciones de IAM que desee utilizar. Le recomendamos que limite sus políticas de IAM al conjunto de permisos más pequeño posible. Sin embargo, puede permitir que su usuario acceda a todas las funciones de IAM, por ejemplo, si utiliza una cuenta de aprendizaje personal con datos de muestra. Para permitir que la lista acceda a todas las funciones de IAM, cambie la lista de muestra por una entrada:. `"arn:aws:iam::111122223333:role/*"`

En la siguiente tabla se describen los permisos que esta política concede.

Action	Resource	Descripción
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Le permite obtener una vista previa de sus archivos.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite enumerar buckets de Amazon S3 de proyectos , conjuntos de datos y trabajos.
`"s3:PutObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permite enviar archivos de salida a S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*",`	Permite eliminar un objeto creado por DataBrew.

Action	Resource	Descripción
	`"arn:aws:s3:::bucket_name"`	
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Permite descifrar conjuntos de datos cifrados.
`"kms:GenerateDataKey*"`	`"arn:aws:kms:::key/key_ids"`	Permite cifrar la salida del trabajo.

Para definir la política de AwsGlueDataBrewS3EncryptedPolicy IAM para DataBrew (consola)

1. Descarga el JSON para la política de `AwsGlueDataBrewS3EncryptedPolicy`IAM.
2. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala en. https://console.aws.amazon.com/iam/
3. En el panel de navegación, seleccione Políticas.
4. Para cada política, elija Crear política.
5. En la pantalla Crear política, vaya a la pestaña JSON.
6. Pegue la declaración JSON de la política sobre la declaración de ejemplo en el editor.
7. Comprueba que la política se adapte a tu cuenta, a los requisitos de seguridad y a AWS los recursos necesarios. Si necesita realizar cambios, puede hacerlos en el editor.
8. Elija Revisar política.

Para definir la política AwsGlueDataBrewS3EncryptedPolicy de IAM para DataBrew (AWS CLI)

1. Descargue el JSON para. `AwsGlueDataBrewS3EncryptedPolicy`
2. Personalice la política como se describe en el primer paso del procedimiento anterior.
3. Ejecute el siguiente comando para crear la política.

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

Añadir usuarios o grupos con DataBrew permisos

Para administrar los permisos, debe asignar políticas a las funciones y funciones a los usuarios y grupos. Para obtener más información, consulte Identidades de IAM (usuarios, grupos y roles) en la Guía del usuario de IAM.

Antes de empezar, debe tener al menos un usuario al que asignar permisos.

Utilice el siguiente procedimiento para configurar DataBrew los permisos para los usuarios que necesitan trabajar en la DataBrew consola o ejecutar DataBrew comandos en la CLI.

Para configurar DataBrew los permisos

1. Cree una clave de acceso para que el usuario utilice el AWS CLI formulario DataBrew y otras herramientas de desarrollo.
2. Habilite el Consola de administración de AWS acceso para que el usuario pueda usar la AWS consola.
3. Cree un rol para DataBrew los usuarios o grupos.
4. Elija la política que va a utilizar. Realice una de las siguientes acciones:
   - Si la creó`AwsGlueDataBrewCustomUserPolicy`, selecciónela de la lista.
   - Para usar la AWS-managed política, selecciónela `AwsGlueDataBrewFullAccessPolicy` de la lista.
5. Asigne esa política al rol.
6. Establezca las relaciones de confianza para el rol de modo que un usuario o grupo pueda asumir el rol correspondiente.
   - Si no utiliza grupos, confíe el rol al usuario.
   - Si utiliza grupos, confíe el rol al grupo y añada el usuario al grupo.

Añadir un rol de IAM con permisos de recursos de datos

Las funciones de IAM se utilizan para gestionar las políticas que se asignan juntas. Un rol de IAM lo puede usar alguien que desempeñe un rol concreto, como un DataBrew usuario o DataBrew él mismo. Para obtener más información, consulte Roles de IAM en la Guía del usuario de IAM.

Utilice el siguiente procedimiento para crear un rol de IAM que sea necesario para que los DataBrew proyectos accedan a los datos.

Para adjuntar la política de IAM requerida a una nueva función de IAM para DataBrew

1. En el panel de navegación, elija Roles, Crear rol.
2. En Seleccione el tipo de entidad de confianza, elija la tarjeta etiquetada como AWS servicio.
3. Elija una opción DataBrewde la lista y, a continuación, elija Siguiente: permisos.
4. Introduzca **`AwsGlueDataBrewDataResourcePolicy`** en el cuadro de búsqueda (la política de IAM que creó en un paso anterior). Seleccione la política y elija Siguiente: etiquetas.
5. Elija Siguiente: Revisar.
6. En Nombre del rol, ingrese **`AwsGlueDataBrewDataAccessRole`** y, luego, elija Crear rol.

Configuración AWS IAM Identity Center(Centro de identidad de IAM)

Con el AWS IAM Identity Center(Centro de identidad de IAM), sus usuarios pueden iniciar sesión DataBrew con una URL simple, sin necesidad de iniciar sesión Consola de administración de AWS y sin necesidad de una AWS cuenta.

Para configurar el Centro de Identidad de IAM

1. Abra la AWS Organizations consola y cree una organización si aún no la tiene. Todas las funciones están habilitadas de forma predeterminada para esta organización.

   Para obtener más información, consulte AWS IAM Identity Center Requisitos previos y Creación y administración de una organización.
2. Abra la consola de AWS IAM Identity Center.
3. Seleccione el origen de la identidad.

   De forma predeterminada, obtiene un almacén de IAM Identity Center para administrar los usuarios de forma rápida y sencilla. Si lo prefiere, puede conectar un proveedor de identidad externo o conectar un AWS Managed Microsoft AD directorio con su Active Directory local. En esta guía, utilizamos el almacén predeterminado del Centro de identidades de IAM.

   Para obtener más información, consulte Elija su fuente de identidad en la Guía del AWS IAM Identity Center usuario.
4. Cree un conjunto de permisos de DataBrew acceso:

a. En el panel de navegación del IAM Identity Center, seleccione AWS cuentas y, a continuación, seleccione Conjuntos de permisos.

b. En la página Crear conjunto de permisos, seleccione Crear un conjunto de permisos personalizado.

c. En Estado de retransmisión, introduzca`https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`.

Introducirlo permite a los usuarios ir directamente a DataBrew.

d. Elija Adjuntar políticas AWS gestionadas DataBrew, busque y elija AwsGlueDataBrewFullAccessPolicy. Al elegir esta opción, los usuarios tendrán todos los permisos que necesitan DataBrew. Puede encontrar más información en[Añadir una política de IAM para un usuario de consola](#).

e. (Opcional) Elige Crear una política de permisos personalizada y personaliza los permisos para tus usuarios.

5. En el panel de navegación de IAM Identity Center, seleccione Grupos y, a continuación, seleccione Crear un grupo. Escriba el nombre del grupo y elija Crear.

6. Añada un usuario a la tienda del IAM Identity Center:

a. En el panel de navegación de IAM Identity Center, elija Usuarios.

b. En la pantalla Añadir usuario, introduzca la información necesaria y seleccione Enviar un correo electrónico al usuario con las instrucciones de configuración de la contraseña. El usuario debería recibir un correo electrónico con los siguientes pasos de configuración.

c. Elija Siguiente: Grupos, el grupo que desea y Añadir usuario.

Los usuarios deberían recibir un correo electrónico en el que se les invite a usar el SSO. En este correo electrónico, deben elegir Aceptar la invitación y establecer la contraseña. También pueden encontrar la URL del portal en el correo electrónico. Pueden usar esta URL para acceder DataBrew.

7. Asigne a cada usuario una cuenta:

a. Abra la [consola del IAM Identity Center](#) y, en el panel de navegación, seleccione AWS cuentas.

b. Elija AWS la organización y elija una AWS cuenta.

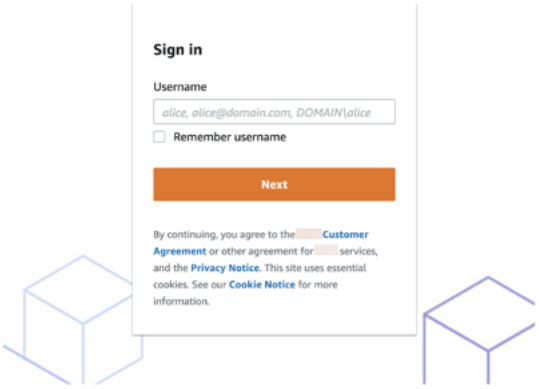
c. En la pantalla Asignar usuarios, seleccione la pestaña Grupos y elija el grupo que desee.

d. Elija Next: Permissions sets (Siguiente: conjuntos de permisos).

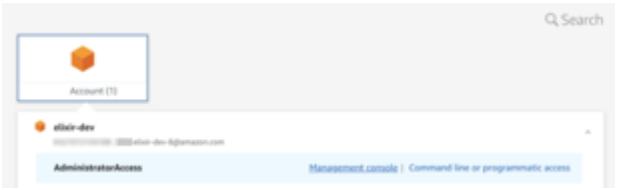
e. Elija el conjunto de permisos DataBrew y elija Finalizar.

Pasos de inicio de sesión para un usuario de IAM Identity Center-enabled

1. Inicie sesión AWS con una cuenta de identidad Center-enabled de IAM.



2. Haga clic en Identidad de la AWS cuenta



3. Haz clic en la consola de administración para redirigirte a la consola con un solo clic. DataBrew

DataBrew Utilización como extensión en JupyterLab

> **Warning**
>
> AWS Glue DataBrew JupyterLab El soporte de extensión finalizará el 31 de diciembre de 2024, ya que JupyterLab 3 personas llegarán a su fin. Para obtener más información, consulte el artículo JupyterLab 3 sobre el fin del mantenimiento.

Si prefiere preparar los datos en un entorno de Jupyter Notebook, puede utilizar todas las funciones del AWS Glue DataBrew mismo. JupyterLab

JupyterLab es un entorno de desarrollo interactivo basado en la web para Jupyter Notebook. En la JupyterLab página web local, puedes añadir secciones para una terminal, una sesión de SQL, Python y mucho más. Tras instalar la AWS Glue DataBrew extensión, puedes añadir una sección

para la DataBrew consola. Se ejecuta con cualquier bloc de notas existente u otras extensiones que ya tenga, directamente desde el JupyterLab entorno.

Temas

Requisitos previos

Antes de empezar, configure los siguientes elementos:

- Una AWS cuenta: si aún no tienes una, empieza conConfigurar un nuevo AWS inscrita.
- Un usuario AWS Identity and Access Management(de IAM) con acceso a los permisos necesarios para DataBrew : para obtener más información, consulteAñadir usuarios o grupos con DataBrew permisos.
- Un rol de IAM para usar en DataBrew las operaciones: puede usar el predeterminado, si `AwsGlueDataBrewDataAccessRole` está configurado. Para configurar funciones de IAM adicionales, consulte. Añadir un rol de IAM con permisos de recursos de datos
- Una JupyterLab instalación (versión 2.2.6 o superior): para obtener más información, consulte los siguientes temas de la JupyterLab documentación:
  - JupyterLab Requisitos previos
  - JupyterLab instalación: se recomienda utilizar`pip install jupyterlab`.
- Una Node.js instalación (versión 12.0 o superior).
- Una AWS Command Line Interface(AWS CLI) instalación: para obtener más información, consulteConfiguración del AWS CLI.
- Una instalación proxy de AWS Jupyter (`pip install aws-jupyter-proxy`): esta extensión se utiliza con un terminal de AWS servicio para transferir sus AWS credenciales de forma segura. Para obtener más información, consulte aws-jupyter-proxy en. GitHub

Para comprobar que tiene instalados los requisitos previos, puede ejecutar una prueba similar a la siguiente en la línea de comandos, como se muestra en el siguiente ejemplo.

```
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
```

El resultado debería tener un aspecto similar al siguiente. Los directorios varían según el sistema operativo y la configuración.

```
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
```

Configuración JupyterLab para usar la extensión

Tras la instalación JupyterLab, debe configurarla para proteger el acceso a los datos y habilitar las extensiones de servidor.

Para configurar una contraseña y un cifrado

1. Establezca una contraseña para proteger los datos que va a añadir a la extensión. Jupyter proporciona una utilidad de contraseñas. Ejecute el siguiente comando y escriba la contraseña que desee en el símbolo del sistema.

   ```
   jupyter notebook password
   ```

   El resultado es similar al siguiente.

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

2. Habilite el cifrado en el servidor Jupyter. Si instalas Jupyter en tu máquina local y nadie puede acceder a ella a través de la red, puedes saltarte este paso.

   Para configurar el cifrado con Transport Layer Security (TLS), cree un certificado personalizado para su entorno. Para obtener más información, consulte la documentación de Jupyter sobre cómo usar Let's Encrypt para proteger un servidor.

3. Para empezar JupyterLab, ejecute el siguiente comando en la línea de comandos.

   ```
   jupyter lab
   ```

   Para obtener más información, consulte Empezar JupyterLab en la JupyterLab documentación.

4. Mientras JupyterLab se ejecuta, puede acceder a él en una URL similar a la siguiente: http://localhost:8888/lab. Si configuras el cifrado, úsalo `https` en lugar de`http`. Si ha personalizado el puerto, sustitúyalo por su número de puerto por`8888`.

Utilice el siguiente procedimiento para habilitar las extensiones de terceros.

Para habilitar extensiones de terceros en JupyterLab

1. En la JupyterLab página web, selecciona el icono del administrador de extensiones en el menú de la izquierda.
2. Lee la advertencia sobre los riesgos de ejecutar extensiones de terceros. Instala únicamente extensiones de desarrolladores en los que confíes.

3. Para activar las extensiones de terceros JupyterLab, selecciona Activar.
4. Sigue las instrucciones para reconstruir y volver JupyterLab a cargar.

Habilitar la extensión para DataBrew JupyterLab

Tras realizar una instalación segura JupyterLab con las extensiones habilitadas, instale la DataBrew extensión para que pueda ejecutarla DataBrew en su ordenador portátil.

Para instalar las extensiones para DataBrew (consola)

1. Para empezar JupyterLab, ejecute el siguiente comando en la línea de comandos.

   ```
   jupyter lab
   ```

2. En la JupyterLab página web, selecciona el icono de Extension Manager en el menú de la izquierda.
3. Busca la DataBrew extensión introduciendo «**brew**» en la esquina superior izquierda para buscar.
4. Busque aws_glue_databrew_jupyter en la lista, pero no haga clic en él. Si hace clic en el nombre resaltado de la extensión, se abrirá una nueva ventana del navegador con la página aws_glue_databrew_jupyter activada. GitHub
5. Para instalar la extensión, elija una de las siguientes opciones: DataBrew

   - En la línea de comandos, ejecute`jupyter labextension install aws_glue_databrew_jupyter`.
   - Seleccione Instalar en la parte inferior de la tarjeta de extensión, debajo de «aws_glue_databrew_jupyter» en letras grises.

   DataBrew la JupyterLab extensión es compatible con las versiones 1.2 y 2.x.
6. Para comprobar que está instalada, ejecute`jupyter labextension list`. El resultado debería tener un aspecto similar al siguiente.

   ```
   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. Realice JupyterLab la reconstrucción mediante una de las siguientes opciones:

   - En la línea de comandos, ejecute`jupyter lab build`.
   - En la página web, selecciona Reconstruir en la parte superior izquierda.

8. Cuando se complete la compilación, realice una de las siguientes acciones:

   - En la línea de comandos, ejecute`jupyter lab`.
   - En la página web, selecciona Recargar en el mensaje Build Complete.

9. En la JupyterLab página web, cierra el Administrador de extensiones seleccionando su icono en el menú de la izquierda.

   Para abrir la extensión, selecciona Iniciar en la sección Otros AWS Glue DataBrew de la pestaña Lanzador. La extensión usa tu AWS CLI configuración actual para las claves de acceso y los ajustes AWS regionales.

Una vez completada la configuración, puede utilizar la AWS Glue DataBrewpestaña para interactuar con ella DataBrew desde dentro JupyterLab.

Introducción al AWS Glue DataBrew

Puede utilizar el siguiente tutorial como guía en la creación de su primer DataBrew proyecto. Carga un conjunto de datos de muestra, ejecuta transformaciones en ese conjunto de datos, crea una receta para capturar esas transformaciones y ejecuta un trabajo para escribir los datos transformados en Amazon S3.

Temas

Requisitos previos

Antes de continuar, siga las instrucciones correspondientes que se indican enConfiguración AWS Glue DataBrew. A continuación, continúePaso 1: Crear un proyecto.

Paso 1: Crear un proyecto

En este paso, utilizará la DataBrew consola para empezar rápidamente con un proyecto de ejemplo.

Para crear un proyecto

1. Inicie sesión en Consola de administración de AWS y abra la DataBrew consola en https://console.aws.amazon.com/databrew/.
2. Asegúrese de que su AWS región esté seleccionada en la parte superior derecha de la DataBrew consola. Para ver una lista de AWS las regiones admitidas por DataBrew, consulta los DataBrew puntos finales y las cuotas en.Referencia general de AWS
3. En el panel de navegación, elija Proyectos y, a continuación, elija Crear proyecto.

4. En el panel de detalles del proyecto, haga lo siguiente:

   - En Nombre del proyecto, introduzca`chess-project`.
   - En Receta adjunta, cree una receta nueva. Se proporciona un nombre sugerido para la receta (`chess-project-recipe`).

5. En el panel Seleccione un conjunto de datos, elija Archivos de muestra.
6. En el panel Archivos de muestra, selecciona Movimientos famosos de una partida de ajedrez. Este conjunto de datos contiene información detallada sobre más de 20 000 partidas de ajedrez.

   Para el nombre del conjunto de datos, se proporciona un nombre sugerido para el conjunto de datos (`chess-games`).
7. En el panel de permisos de acceso, elija`AwsGlueDataBrewDataAccessRole`. Se trata de una función vinculada a un servicio que le permite DataBrew acceder a sus buckets de Amazon S3 en su nombre.
8. Elija Crear proyecto y espere hasta que DataBrew termine de preparar el proyecto. La ventana tiene un aspecto similar al siguiente.

   Los datos que ve representan una muestra del `chess-games` conjunto de datos. De forma predeterminada, la muestra consta de las primeras 500 filas del conjunto de datos. Puede cambiar la configuración de este proyecto más adelante.

   La barra de herramientas proporciona acceso a cientos de transformaciones de datos que puede aplicar a los datos.

   El panel de recetas situado a la derecha de la DataBrew consola hace un seguimiento de las transformaciones que ha aplicado hasta el momento.

Paso 2: resumir los datos

En este paso, se crea una DataBrew receta: un conjunto de transformaciones que se pueden aplicar a este conjunto de datos y a otros similares. Cuando la receta esté completa, la publicas para que esté disponible para su uso.

En el ajedrez, se puede calificar a los jugadores en función de su desempeño contra otros jugadores. (Para obtener más información, consulte https://en.wikipedia.org/wiki/Chess_rating_system). En este tutorial, te centrarás únicamente en las partidas en las que ambos jugadores eran de clase A, lo que significa que sus puntuaciones eran de 1800 o más.

Para resumir los datos

1. En la barra de herramientas de transformación, elija Filtrar, Por condición, Mayor o igual a.
2. Defina estas opciones de la siguiente manera:

   - Columna de origen - `white_rating`
   - Estado del filtro: mayor o igual a 1800

   Para ver cómo funciona la transformación, selecciona Vista previa de los cambios. A continuación, elija Aplicar.
3. Repite el paso anterior, pero esta vez establece la columna Fuente en`black_rating`. Tras aplicar los cambios, los datos de muestra solo incluyen los juegos en los que los jugadores de cada bando (blanco y negro) eran de clase A o superior.
4. Resuma los datos para determinar cuántas partidas ganó cada bando. Para ello, en la barra de herramientas de transformación, selecciona Agrupar.
5. Para las propiedades del grupo, haga lo siguiente:

   a. En la primera fila, elija el nombre **`winner`** de la columna. Deje la opción Agregar establecida en Agrupar por.
   b. En la segunda fila, elija el nombre **`victory_status`** de la columna. Deje la opción Agregar establecida en Agrupar por.
   c. Seleccione Añadir otra columna.
   d. En la tercera fila, elija el nombre **`winner`** de la columna. Defina Agregar como Recuento.
   e. En Tipo de grupo, elija Agrupar como tabla nueva. El panel de vista previa muestra el aspecto que tendrá el resultado.
   f. Seleccione Finalizar.
6. Elija Publicar para guardar su trabajo, a la derecha del panel de recetas.
7. En Descripción de la versión, ingresa Primera versión de mi receta. Luego elige Publicar.

Paso 3: Añadir más transformaciones

En este paso, añades más transformaciones a tu receta y publicas otra versión de la misma. Para afinar nuestro ejemplo, utilizamos la información de que no todas las partidas de ajedrez dan como resultado un ganador claro; algunas partidas se juegan en empate.

Para añadir más transformaciones a las recetas y volver a publicarlas

1. En la barra de herramientas de transformación, selecciona Filtrar, Por condición, Es para no eliminar las partidas que se jugaron en tablas.
2. Configure estas opciones de la siguiente manera:

   - Columna de origen - `victory_status`
   - Estado del filtro: no lo es `draw`

   Para añadir esta transformación a tu receta, selecciona Aplicar.
3. Cambia los datos `victory_status` para que sean más significativos. Para ello, en la barra de herramientas de transformación, elija Limpiar, Reemplazar, Reemplazar valor o patrón.
4. Defina estas opciones de la siguiente manera:

   - Columna de origen - `victory_status`
   - Especifique los valores que desea reemplazar: valor o patrón
   - Valor que se va a reemplazar - `mate`
   - Sustituir por el valor - `checkmate`

   Para añadir esta transformación a la receta, selecciona Aplicar.
5. Repite el paso anterior, pero cambia `resign` a`other player resigned`.
6. Repita el paso anterior, pero cambie `outoftime` a`time ran out`.
7. Selecciona Publicar para guardar tu trabajo, a la derecha del panel de recetas.

Paso 4: Revisa tus DataBrew recursos

Ahora que ha trabajado con un proyecto de muestra, revise los DataBrew recursos que ha creado hasta ahora.

Para revisar tus DataBrew recursos

1. En el panel de navegación, selecciona Conjuntos de datos.

   Cuando creaste el proyecto de muestra, DataBrew creaste un conjunto de datos para ti (`chess-games`). El archivo de datos de origen se almacena en Amazon S3 y está en formato Microsoft

Excel (`chess-games.xlsx`). El archivo contiene metadatos de más de 20 000 partidas de ajedrez. El `chess-games` conjunto de datos proporciona la información DataBrew necesaria para leer los datos de ese archivo.

2. En el panel de navegación, selecciona Proyectos.

   Debería ver el proyecto con el que ha trabajado en los pasos anteriores (`chess-project`). En este caso, cada proyecto requiere un conjunto de datos`chess-games`. Cada proyecto también requiere una receta, de modo que puedas añadir pasos de transformación de datos a medida que avanzas. Cuando creó este proyecto de muestra, DataBrew creó una nueva receta (vacía) para usted y la adjuntó al proyecto.
3. En el panel de navegación, elija Recetas y, en la columna Nombre de la receta, elija chess-project-recipe. Aquí se muestra la receta que se DataBrew creó para el proyecto y que se ha perfeccionado añadiéndole pasos de transformación.
4. A la izquierda, consulta las versiones de recetas que se han publicado. Elige una de estas opciones para ver la pestaña de pasos de la receta, que muestra los detalles de la receta y los pasos de esa versión.
5. Consulta la pestaña de linaje de datos, que muestra de dónde provienen los datos y cómo se utilizan. Para obtener más información, elige cualquiera de los íconos del diagrama.

Paso 5: Cree un perfil de datos

Cuando trabaja en un proyecto, DataBrew muestra estadísticas como el número de filas de la muestra y la distribución de valores únicos en cada columna. Estas estadísticas, y muchas más, representan un perfil de la muestra.

Para solicitar un perfil de datos, cree y ejecute un trabajo de perfil.

Para perfilar un conjunto de datos

1. En el panel de navegación, elija Trabajos.
2. En la pestaña Trabajos de perfil, elija Crear trabajo.
3. En Nombre del trabajo, introduzca`chess-data-profile`.
4. En Tipo de trabajo, elija Crear un trabajo de perfil.
5. En el panel de entrada de Job, haga lo siguiente:

   - En Ejecutar, elija Conjunto de datos.

- Elija Seleccionar un conjunto de datos para ver una lista de los conjuntos de datos disponibles y elija`chess-games`.

6. En el panel de configuración de salida de Job, haga lo siguiente:

   - En Tipo de archivo, elija JSON (notación de JavaScript objetos).
   - Elija la ubicación de S3 para ver una lista de los depósitos de Amazon S3 disponibles y elija el depósito que desee utilizar. A continuación, seleccione Examinar. En la lista de carpetas`databrew-output`, elija y elija Seleccionar.

7. En el panel de permisos de acceso, elija`AwsGlueDataBrewDataAccessRole`. Se trata de un rol vinculado a un servicio que le permite DataBrew acceder a sus buckets de Amazon S3 en su nombre.
8. Elija Crear y ejecutar trabajo. DataBrew crea un trabajo con su configuración y, a continuación, lo ejecuta.
9. En el panel Historial de ejecuciones de trabajos, espere a que el estado del trabajo cambie de `Running` a`Succeeded`.
10. Para ver el perfil, seleccione VER PERFIL:

    Aparece la ventana CONJUNTOS DE DATOS. Tómate un tiempo para explorar las siguientes pestañas:

    - Vista previa del conjunto de
    - Descripción general del perfil
    - Estadísticas de las columnas
    - Estadísticas de linaje de datos

Paso 6: Transformar el conjunto de datos

Hasta ahora, probaste tu receta solo en una muestra del conjunto de datos. Ahora es el momento de transformar todo el conjunto de datos mediante la creación de una DataBrew receta.

Cuando se ejecute el trabajo, DataBrew aplica la receta a todos los datos del conjunto de datos y escribe los datos transformados en un bucket de Amazon S3. Los datos transformados están separados del conjunto de datos original. DataBrew no altera los datos de origen.

Antes de continuar, asegúrate de tener un bucket de Amazon S3 en tu cuenta al que puedas escribir. En ese depósito, cree una carpeta desde la que capturar el resultado del trabajo DataBrew. Para realizar estos pasos, utilice el siguiente procedimiento.

Para crear un depósito y una carpeta de S3 para capturar los resultados del trabajo

1. Inicie sesión en la consola de Amazon S3 Consola de administración de AWS y ábrala en https://console.aws.amazon.com/databrew/.

   Si ya tiene un bucket de Amazon S3 disponible y tiene permisos de escritura para él, omita el paso siguiente.
2. Si no tiene un bucket de Amazon S3, elija Create bucket. En el campo Nombre del bucket, introduce un nombre exclusivo para el nuevo bucket. Elija Crear bucket.
3. En la lista de cubos, elige el que quieras usar.
4. Elija Crear carpeta.
5. En Nombre de carpeta`databrew-output`, introduzca y seleccione Crear carpeta.

Tras crear un bucket y una carpeta de Amazon S3 para contener el trabajo, ejecútelo mediante el siguiente procedimiento.

Para crear y ejecutar un trabajo de receta

1. En el panel de navegación, elija Trabajos.
2. En la pestaña Trabajos de recetas, elija Crear trabajo.
3. En Nombre del trabajo, introduzca`chess-winner-summary`.
4. En Tipo de trabajo, selecciona Crear un trabajo de receta.
5. En el panel de entrada de Job, haga lo siguiente:

   - En Ejecutar, elija Conjunto de datos.
   - Elija Seleccionar un conjunto de datos para ver una lista de los conjuntos de datos disponibles y elija`chess-games`.
   - Elija Seleccione una receta para ver una lista de las recetas disponibles y elija`chess-project-recipe`.
6. En el panel de configuración de salida de Job, haga lo siguiente:

   - Tipo de archivo: elija CSV (valores separados por comas).

- Ubicación de S3: seleccione este campo para ver una lista de los depósitos de Amazon S3 disponibles y elija el depósito que desee utilizar. A continuación, seleccione Examinar. En la lista de carpetas`databrew-output`, selecciona y selecciona Seleccionar.

7. En el panel de permisos de acceso, elija`AwsGlueDataBrewDataAccessRole`. Este rol vinculado a un servicio le permite DataBrew acceder a sus buckets de Amazon S3 en su nombre.
8. Elija Crear y ejecutar trabajo. DataBrew crea un trabajo con su configuración y, a continuación, lo ejecuta.
9. En el panel Historial de ejecuciones de trabajos, espere a que el estado del trabajo cambie de `Running` a`Succeeded`.
10. Elija Output para acceder a la consola Amazon S3. Elija su bucket de S3 y, a continuación, elija la `databrew-output` carpeta para acceder a la salida del trabajo.
11. (Opcional) Seleccione Descargar para descargar el archivo y ver su contenido.

Paso 7: (opcional) sanear

El tutorial está completo. Puede seguir utilizando los recursos DataBrew y los de Amazon S3 que creó o eliminarlos.

Cómo limpiar los recursos

1. Abra la DataBrew consola en y https://console.aws.amazon.com/databrew/, en el panel de navegación, seleccione Proyectos.
2. Elija su proyecto (proyecto de muestra). En Actions (Acciones), seleccione Delete (Eliminar).
3. En el panel Eliminar un proyecto de muestra, selecciona Eliminar la receta adjunta. A continuación, elija Eliminar. Su proyecto, junto con su receta y sus trabajos, se eliminarán.
4. En el panel de navegación, selecciona Conjuntos de datos.
5. Elija su conjunto de datos (`chess-games`) y, en Acciones, elija Eliminar.
6. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/. Elimina la `databrew-output` carpeta y su contenido.

   (Opcional) Si estás seguro de que ya no necesitas tu bucket de Amazon S3, puedes eliminarlo.

Conectarse a los datos con AWS Glue DataBrew

En AWS Glue DataBrew, un conjunto de datos representa datos que se cargan desde un archivo o se almacenan en otro lugar. Por ejemplo, los datos se pueden almacenar en Amazon S3, en una fuente de datos JDBC compatible o en un catálogo de AWS Glue datos. Si no va a cargar un archivo directamente a DataBrew, el conjunto de datos también contiene detalles sobre cómo DataBrew conectarse a los datos.

Al crear el conjunto de datos (por ejemplo,`inventory-dataset`), se introducen los detalles de la conexión solo una vez. Desde ese punto, DataBrew puede acceder a los datos subyacentes por usted. Con este enfoque, puede crear proyectos y desarrollar transformaciones para sus datos, sin tener que preocuparse por los detalles de conexión o los formatos de archivo.

Temas

Tipos de archivos compatibles para las fuentes de datos

Los siguientes requisitos de archivos se aplican a los archivos almacenados en Amazon S3 y a los archivos que carga desde una unidad local. DataBrew admite los siguientes formatos de archivo: valores separados por comas (CSV), Microsoft Excel, JSON, ORC y Parquet. Puede utilizar archivos con una extensión no estándar o sin extensión si el archivo es de uno de los tipos admitidos.

Si DataBrew no puede deducir el tipo de archivo, asegúrese de seleccionar usted mismo el tipo de archivo correcto (CSV, Excel, JSON, ORC o Parquet). Se admiten los archivos CSV, JSON, ORC y Parquet comprimidos, pero los archivos CSV y JSON deben incluir el códec de compresión como extensión del archivo. Si va a importar una carpeta, todos los archivos de la carpeta deben ser del mismo tipo de archivo.

Los formatos de archivo y los algoritmos de compresión compatibles se muestran en la siguiente tabla.

> **Note**
>
> Los archivos CSV, Excel y JSON deben estar codificados con Unicode (UTF-8).

Formato	Extensión de archivo (opcional)	Extensiones para archivos comprimidos (obligatorias)

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

Libro de trabajo de Microsoft Excel	`.xlsx`	Sin soporte de compresión

JSON (documento JSON y líneas JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

Apache ORC	`.orc`	`.zlib` `.snappy`
Apache Parquet	`.parquet`	`.gz` `.snappy`

Formato	Extensión de archivo (opcional)	Extensiones para archivos comprimidos (obligatorias)
		`.lz4`

Conexiones compatibles para fuentes y salidas de datos

Puede conectarse a las siguientes fuentes de datos para realizar trabajos con DataBrew recetas. Estas incluyen cualquier fuente de datos que no sea un archivo al que vaya a DataBrew subir directamente. La fuente de datos que está utilizando podría denominarse base de datos, almacén de datos o cualquier otro nombre. Nos referimos a todos los proveedores de datos como fuentes de datos o conexiones.

Puede crear un conjunto de datos utilizando cualquiera de las siguientes fuentes de datos.

También puede utilizar las bases de datos Amazon S3 o JDBC compatibles con Amazon RDS para la salida de los trabajos de DataBrew recetas.AWS Glue Data Catalog Amazon AppFlow y AWS Data Exchange no son almacenes de datos compatibles para la producción de trabajos de DataBrew recetas.

- Amazon S3

  Puede usar S3 para almacenar y proteger cualquier cantidad de datos. Para crear un conjunto de datos, especifique una URL de S3 desde la que DataBrew pueda acceder a un archivo de datos, por ejemplo: `s3://your-bucket-name/inventory-data.csv`

  DataBrew también puede leer todos los archivos de una carpeta S3, lo que significa que puede crear un conjunto de datos que abarque varios archivos. Para ello, especifique una URL de S3 de esta forma:`s3://your-bucket-name/your-folder-name/`.

  DataBrew solo admite las siguientes clases de almacenamiento de Amazon S3: Estándar, Redundancia reducida y S3 One Zone-IA. Standard-IA DataBrew ignora los archivos con otras clases de almacenamiento. DataBrew también ignora los archivos vacíos (archivos que contienen 0 bytes). Para obtener más información sobre las clases de almacenamiento de Amazon S3, consulte Uso de las clases de almacenamiento de Amazon S3 en la Guía del usuario de Amazon S3 Console.
- AWS Glue Data Catalog

Puede usar el catálogo de datos para definir las referencias a los datos almacenados en la AWS nube. Con el catálogo de datos, puede crear conexiones a tablas individuales en los siguientes servicios:

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

DataBrew también puede leer todos los archivos de una carpeta de Amazon S3, lo que significa que puede crear un conjunto de datos que abarque varios archivos. Para ello, especifique una URL de Amazon S3 de la siguiente forma: `s3://your-bucket-name/your-folder-name/`

Para poder utilizarlas con DataBrew, las tablas de Amazon S3 definidas en AWS Glue Data Catalog, deben tener agregada una propiedad de tabla llamada `aclassification`, que identifique el formato de los datos como `csv` `jsonparquet`, o y `typeOfData` como`file`. Si la propiedad de la tabla no se agregó cuando se creó la tabla, puede agregarla mediante la AWS Glue consola.

DataBrew solo admite las clases de almacenamiento Standard, Reduced Redundancy y S3 One Zone-IA de Amazon S3. Standard-IA DataBrew ignora los archivos con otras clases de almacenamiento. DataBrew también ignora los archivos vacíos (archivos que contienen 0 bytes). Para obtener más información sobre las clases de almacenamiento de Amazon S3, consulte Uso de las clases de almacenamiento de Amazon S3 en la Guía del usuario de Amazon S3 Console.

DataBrew también puede acceder a las tablas de AWS Glue Data Catalog S3 desde otras cuentas si se crea una política de recursos adecuada. Puede crear una política en la AWS Glue consola, en la pestaña Configuración, en el Catálogo de datos. El siguiente es un ejemplo de política específica para una sola persona Región de AWS.

> **Warning**
>
> Se trata de una política de recursos muy permisiva que concede acceso `*$ACCOUNT_TO*` sin restricciones al catálogo de datos de. `*$ACCOUNT_FROM*` En la mayoría de los casos, se recomienda limitar la política de recursos a catálogos o tablas específicos. Para obtener más información, consulte las políticas AWS Glue de recursos para el control de acceso en la Guía para AWS Glue desarrolladores.

En algunos casos, puede que desee crear un proyecto o ejecutar un trabajo *$ACCOUNT_TO* con una tabla de S3 del catálogo de AWS Glue datos *$ACCOUNT_FROM* que apunte a una ubicación de S3 que también esté en*$ACCOUNT_FROM*.AWS Glue DataBrew En esos casos, la función de IAM utilizada al crear el proyecto y la tarea *$ACCOUNT_TO* debe tener permiso para enumerar y obtener los objetos de esa ubicación de *$ACCOUNT_FROM* S3. Para obtener más información, consulta Cómo conceder acceso a varias cuentas en la Guía AWS Glue para desarrolladores.

- Datos conectados mediante controladores JDBC

  Puede crear un conjunto de datos conectándose a los datos con un controlador JDBC compatible. Para obtener más información, consulte Uso de controladores con AWS Glue DataBrew.

  DataBrew admite oficialmente las siguientes fuentes de datos mediante Java Database Connectivity (JDBC):

  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Amazon Redshift
  - Conector Snowflake para Spark

  Las fuentes de datos se pueden ubicar en cualquier lugar desde el que pueda conectarse a ellas. DataBrew Esta lista incluye solo las conexiones JDBC que hemos probado y, por lo tanto, compatibles.

  Las fuentes de datos de Amazon Redshift y Snowflake Connector for Spark se pueden conectar de cualquiera de las siguientes maneras:

  - Con un nombre de tabla.
  - Con una consulta SQL que abarca varias tablas y operaciones.

  Las consultas SQL se ejecutan al iniciar un proyecto o ejecutar una tarea.

  Para conectarse a datos que requieren un controlador JDBC que no figura en la lista, asegúrese de que el controlador sea compatible con el JDK 8. Para usar el controlador, guárdelo en S3 en un depósito al que pueda acceder a él con su función de IAM. DataBrew A continuación, dirija

el conjunto de datos al archivo del controlador. Para obtener más información, consulte Uso de controladores con AWS Glue DataBrew.

Ejemplo de consulta para un SQL-based conjunto de datos:

```
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
```

Limitaciones del SQL personalizado

Si utiliza una conexión JDBC para acceder a los datos de un DataBrew conjunto de datos, tenga en cuenta lo siguiente:

- AWS Glue DataBrew no valida el SQL personalizado que proporciona como parte de la creación del conjunto de datos. La consulta SQL se ejecutará al iniciar un proyecto o ejecutar un trabajo. DataBrew toma la consulta que proporciona y la pasa al motor de base de datos mediante los controladores JDBC predeterminados o proporcionados.
- Un conjunto de datos creado con una consulta no válida fallará cuando se utilice en un proyecto o trabajo. Valide la consulta antes de crear el conjunto de datos.
- La función Validar SQL solo está disponible para las fuentes de Redshift-based datos de Amazon.
- Si desea utilizar un conjunto de datos en un proyecto, limite el tiempo de ejecución de las consultas SQL a menos de tres minutos para evitar que se agote el tiempo de espera durante la carga del proyecto. Comprueba el tiempo de ejecución de la consulta antes de crear un proyecto.

- Amazon AppFlow

Con Amazon AppFlow, puede transferir datos a Amazon S3 desde aplicaciones de terceros Software-as-a-Service (SaaS) como Salesforce, Zendesk, Slack y. ServiceNow A continuación, puede usar los datos para crear un conjunto de datos. DataBrew

En Amazon AppFlow, creas una conexión y un flujo para transferir datos entre tu aplicación de terceros y una aplicación de destino. Cuando utilices Amazon AppFlow con DataBrew, asegúrate de que la aplicación de AppFlow destino de Amazon sea Amazon S3. Las aplicaciones de AppFlow destino de Amazon distintas de Amazon S3 no aparecen en la DataBrew consola. Para obtener más información sobre la transferencia de datos desde una aplicación de terceros y la creación de AppFlow conexiones y flujos de Amazon, consulta la AppFlow documentación de Amazon.

Si selecciona Conectar un nuevo conjunto de datos en la pestaña Conjuntos de datos DataBrew y hace clic en Amazon AppFlow, verá todos los flujos de Amazon AppFlow que están configurados con Amazon S3 como aplicación de destino. Para usar los datos de un flujo para su conjunto de datos, elija ese flujo.

Al seleccionar Crear flujo, Administrar flujos y Ver detalles de Amazon AppFlow en la DataBrew consola, se abre la AppFlow consola de Amazon para que pueda realizar esas tareas.

Después de crear un conjunto de datos desde Amazon AppFlow, puede ejecutar el flujo y ver los detalles de la última ejecución del flujo al ver los detalles del conjunto de datos o los detalles del trabajo. Al ejecutar el flujo DataBrew, el conjunto de datos se actualiza en S3 y está listo para usarse en DataBrew él.

Al seleccionar un AppFlow flujo de Amazon en la DataBrew consola para crear un conjunto de datos, pueden surgir las siguientes situaciones:

- Los datos no se han agregado: si el activador del flujo es Ejecutar bajo demanda o se ejecuta según lo programado con una transferencia de datos completa, asegúrese de agregar los datos del flujo antes de usarlos para crear un DataBrew conjunto de datos. Al agregar el flujo, se combinan todos los registros del flujo en un solo archivo. Los flujos con el tipo de activación Ejecutar según lo programado con transferencia de datos incremental o Ejecutar según el evento no requieren agregación. Para agregar datos en Amazon AppFlow, selecciona Editar configuración de flujo > Detalles de destino > Configuración adicional > Preferencia de transferencia de datos.
- No se ha ejecutado el flujo: si el estado de ejecución de un flujo es vacío, significa una de las siguientes opciones:
  - Si el desencadenante para ejecutar el flujo es Ejecutar bajo demanda, el flujo aún no se ha ejecutado.

- Si el desencadenante para ejecutar el flujo es Ejecutar por evento, el evento desencadenante aún no se ha producido.
- Si el desencadenante para ejecutar el flujo es Ejecutar según lo programado, aún no se ha producido ninguna ejecución programada.

Antes de crear un conjunto de datos con un flujo, elija Ejecutar flujo para ese flujo.

Para obtener más información, consulta Amazon AppFlow flows en la Guía del AppFlow usuario de Amazon.

- AWS Data Exchange

Puede elegir entre cientos de fuentes de datos de terceros que están disponibles en AWS Data Exchange. Al suscribirse a estas fuentes de datos, obtiene la versión más actualizada de los datos.

Para crear un conjunto de datos, debe especificar el nombre de un producto de AWS Data Exchange datos al que está suscrito y al que tiene derecho a usar.

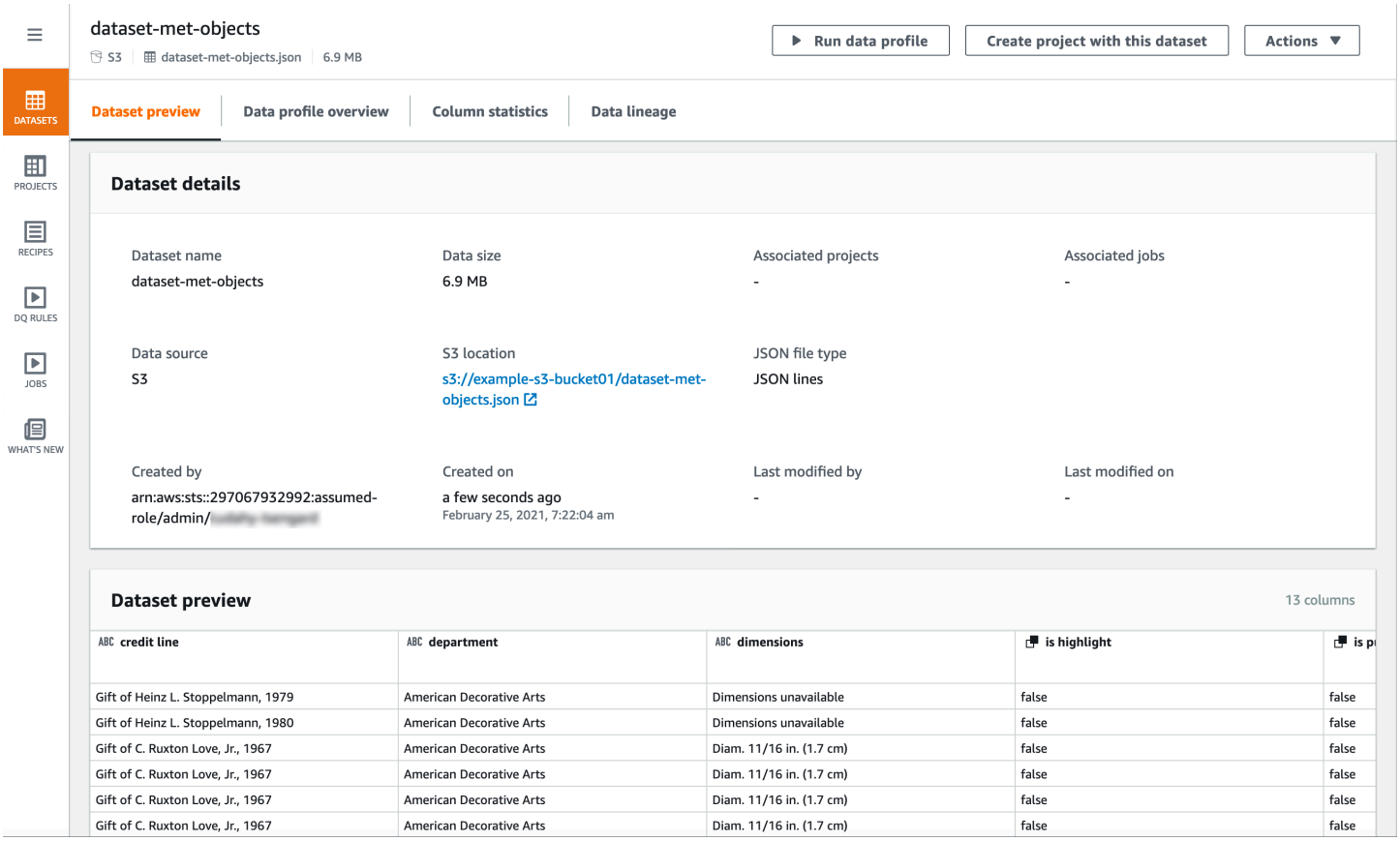
Uso de conjuntos de datos en AWS Glue DataBrew

Para ver una lista de sus conjuntos de datos en la DataBrew consola, elija DATASET a la izquierda. En la página de conjuntos de datos, puedes ver información detallada de cada conjunto de datos haciendo clic en su nombre o seleccionando Acciones y Editar en su menú contextual.

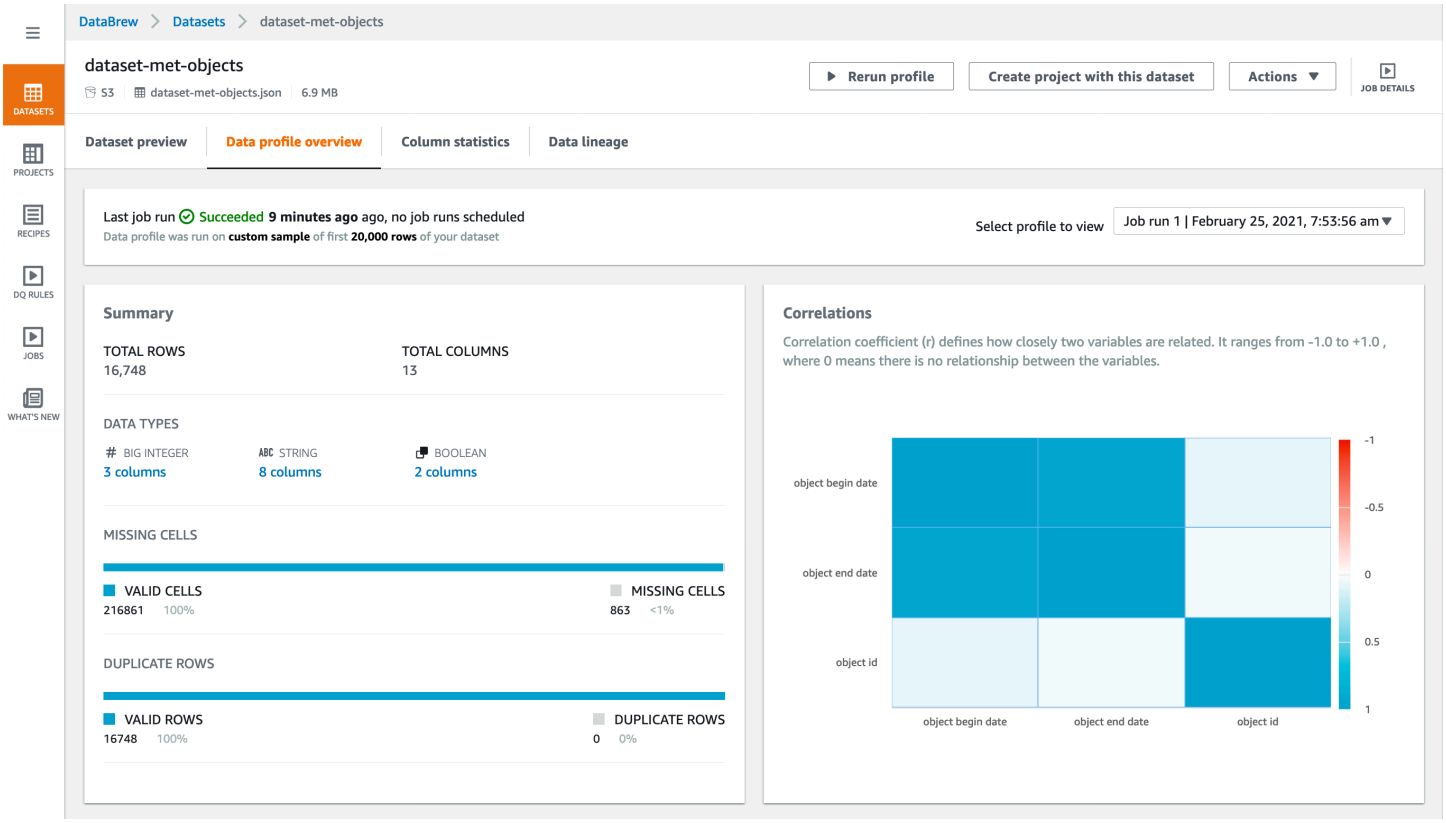
Para crear un nuevo conjunto de datos, elige DATASET, Connect new dataset. Las distintas fuentes de datos tienen distintos parámetros de conexión, y usted los introduce para que DataBrew se puedan conectar. Al guardar la conexión y elegir Crear conjunto de datos, DataBrew se conecta a los datos y comienza a cargarlos. Para obtener más información, consulte Conectarse a tus datos.

La página del conjunto de datos tiene los siguientes elementos para ayudarte a explorar tus datos.

Vista previa del conjunto de datos: en esta pestaña, puede encontrar información de conexión del conjunto de datos y una descripción general de la estructura general del conjunto de datos, como se muestra a continuación.

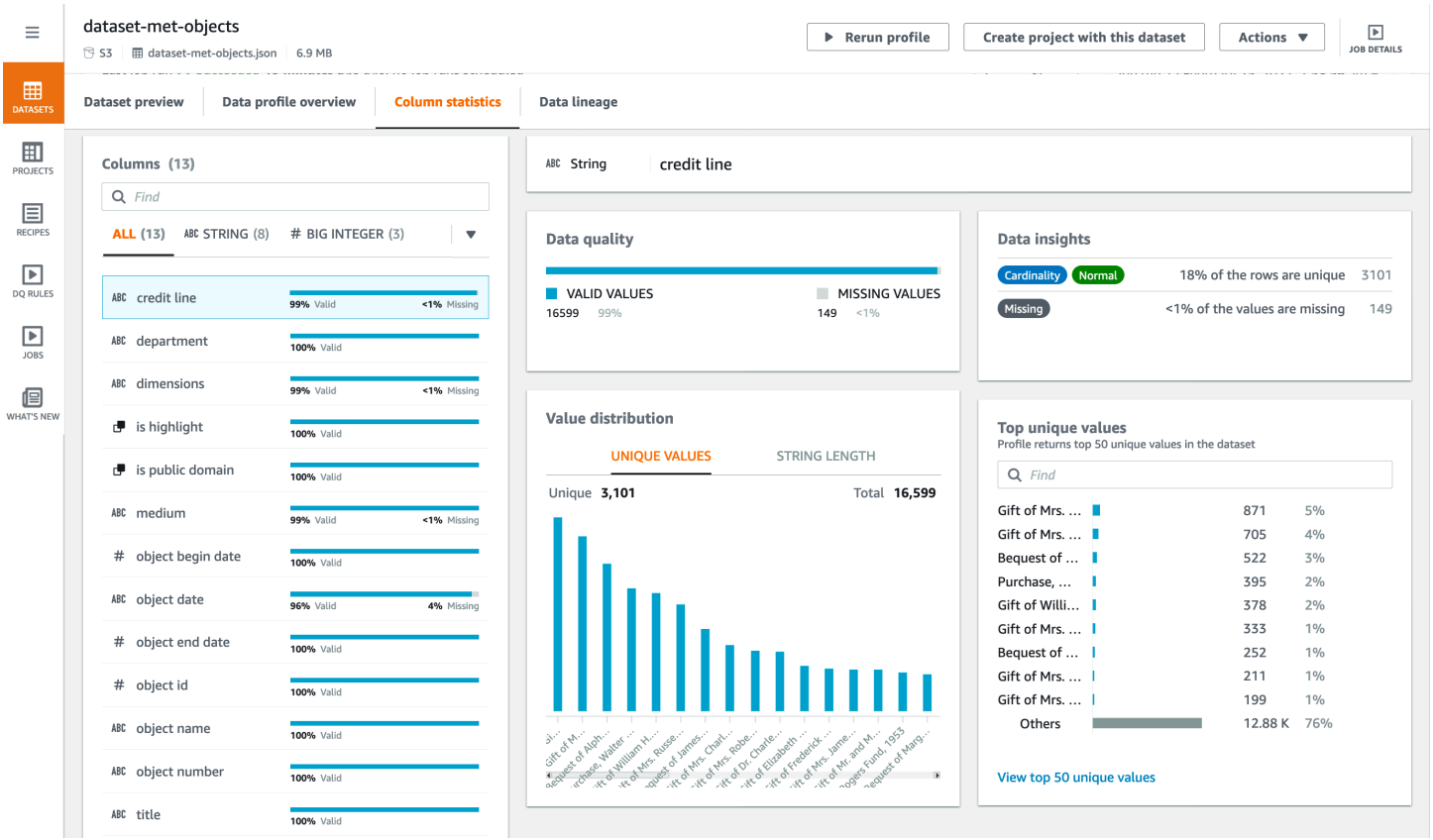


Descripción general del perfil de datos: en esta pestaña, puede encontrar un perfil gráfico de datos estadísticos y volumétricos para su conjunto de datos, como se muestra a continuación.

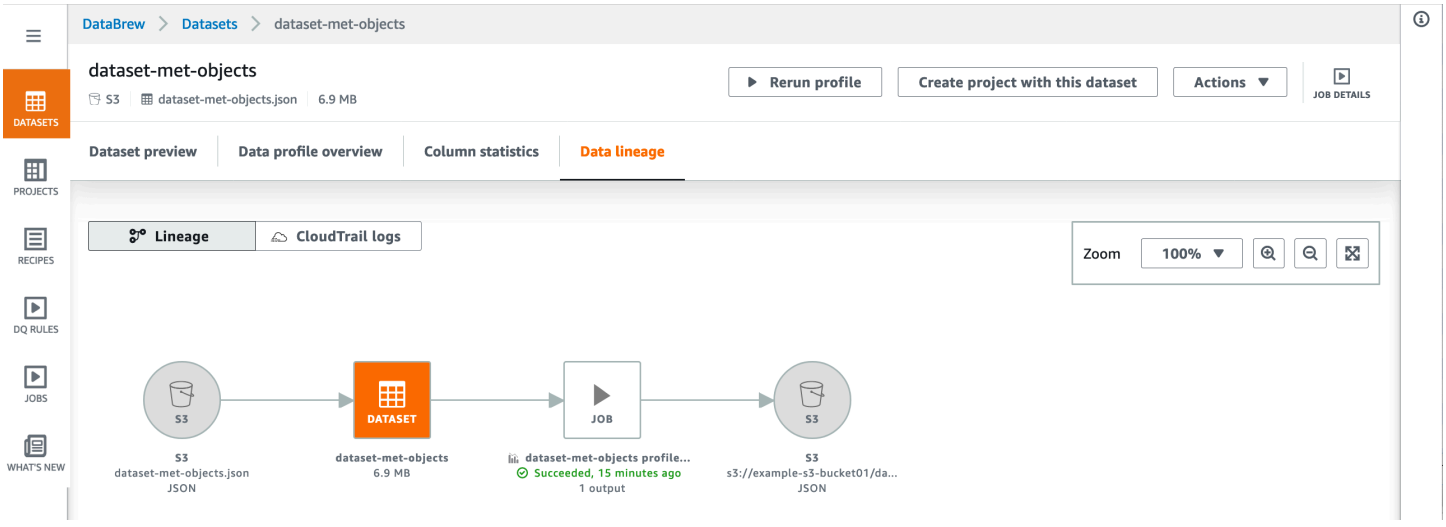


> **Note**
>
> Para crear un perfil de datos, ejecute un trabajo DataBrew de perfil en su conjunto de datos. Para obtener información acerca de cómo hacerlo, consulte Paso 5: Cree un perfil de datos.

Estadísticas de columnas: en esta pestaña, puedes encontrar estadísticas detalladas sobre cada columna de tu conjunto de datos, como se muestra a continuación.



Linaje de datos: esta pestaña muestra una representación gráfica de cómo se creó el conjunto de datos y cómo se utiliza DataBrew, como se muestra a continuación.



Temas

- Eliminación de un conjunto de datos

Eliminación de un conjunto de datos

Si ya no necesitas un conjunto de datos, puedes eliminarlo. La eliminación de un conjunto de datos no afecta en modo alguno a la fuente de datos subyacente. Simplemente elimina la información que DataBrew solía acceder a la fuente de datos.

No puede eliminar un conjunto de datos si otros DataBrew recursos dependen de él. Por ejemplo, si actualmente tienes un DataBrew proyecto que usa el conjunto de datos, borra primero el proyecto antes de eliminar el conjunto de datos.

Para eliminar un conjunto de datos, selecciona Conjunto de datos en el panel de navegación. Elija el conjunto de datos que desea eliminar y, a continuación, en Acciones, elija Eliminar.

Conectarse a tus datos

Para obtener más información sobre cómo conectarse a las siguientes fuentes de datos, elija la sección que corresponda a su caso.

- AWS Glue Data Catalog— Puede utilizar el catálogo de datos para definir las referencias a los objetos de datos almacenados en la AWS nube, incluidos los siguientes servicios:
  - Amazon Redshift
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS para MySQL
  - Amazon RDS para PostgreSQL

  DataBrew reconoce todos los permisos de Lake Formation que se han aplicado a los recursos del catálogo de datos, por lo que DataBrew los usuarios solo pueden acceder a estos recursos si están autorizados.

  Para crear un conjunto de datos, especifique un nombre de base de datos del catálogo de datos y un nombre de tabla. DataBrew se ocupa del resto de los detalles de la conexión.
- AWS Intercambio de datos: puede elegir entre cientos de fuentes de datos de terceros que están disponibles en AWS Data Exchange. Al suscribirse a estas fuentes de datos, siempre dispondrá de la versión más actualizada de los datos.

  Para crear un conjunto de datos, debe especificar el nombre de un producto de datos de Data Exchange al que esté suscrito o al que tenga derecho a usar.

- Conexiones de controladores JDBC: puede crear un conjunto de datos conectándose DataBrew a una JDBC-compatible fuente de datos. DataBrew admite la conexión a las siguientes fuentes a través de JDBC:
  - Amazon Redshift
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Snowflake

Temas

Uso de controladores con AWS Glue DataBrew

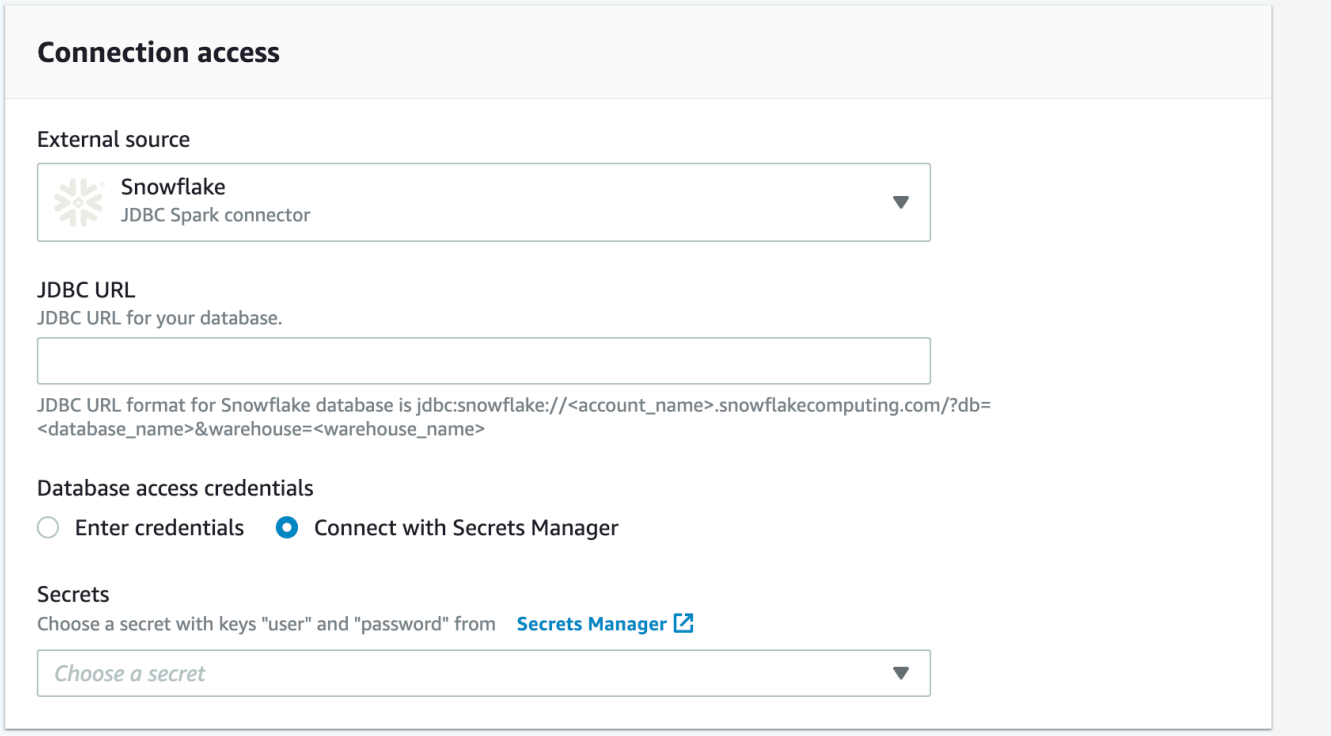
Un controlador de base de datos es un archivo o una URL que implementa un protocolo de conexión a bases de datos, por ejemplo, Java Database Connectivity (JDBC). El controlador funciona como un adaptador o un traductor entre un sistema de administración de bases de datos (DBMS) específico y otro sistema.

En este caso, permite conectarse AWS Glue DataBrew a sus datos. A continuación, puede acceder a un objeto de base de datos, como una tabla o una vista, desde una fuente de datos compatible. La fuente de datos que está utilizando podría denominarse base de datos, almacén de datos o cualquier otro nombre. Sin embargo, a los efectos de esta documentación, nos referimos a todos los proveedores de datos como fuentes de datos o conexiones.

Para usar un controlador JDBC o un archivo jar, descargue el archivo o los archivos que necesite y colóquelos en un depósito de S3. La función de IAM que utilice para acceder a los datos debe tener permisos de lectura para ambos archivos del controlador.

> **Note**
>
> With AWS Glue4.0, la conexión a Snowflake como fuente de datos es compatible de forma nativa. No es necesario que proporcione archivos personalizados. `jar` En AWS Glue

DataBrew, elige Snowflake como conexión de origen externo y proporciona la URL de tu instancia de Snowflake. La URL utilizará un nombre de host en el formulario `https://account_identifier.snowflakecomputing.com`.
Proporcione las credenciales de acceso a los datos, el nombre de la base de datos de Snowflake y el nombre del esquema de Snowflake. Además, si su usuario de Snowflake no tiene un conjunto de almacenes predeterminado, tendrá que proporcionar un nombre de almacén.
Las conexiones de Snowflake utilizan un AWS Secrets Manager secreto para proporcionar información sobre las credenciales. Sus funciones de proyecto y de trabajo deben tener permiso para leer este secreto.



Para usar los controladores con DataBrew

1. Averigüe en qué versión de la fuente de datos se encuentra utilizando el método que proporciona el producto.
2. Busque la última versión de los conectores y el controlador necesarios. Puede encontrar esta información en el sitio web de los proveedores de datos.
3. Descargue la versión requerida de los archivos JDBC. Por lo general, se almacenan como archivos Java ArchiVes (.JAR).

4. Cargue los controladores de la consola a su bucket de S3 o proporcione la ruta de S3 a sus archivos.JAR.
5. Introduzca los detalles básicos de la conexión, por ejemplo, la clase, la instancia, etc.
6. Introduzca cualquier información de configuración adicional que necesite la fuente de datos, por ejemplo, información sobre la nube privada virtual (VPC).

Controladores JDBC compatibles

Producto	Versión de compatible	Instrucciones y descargas del controlador	Se admiten consultas SQL
Microsoft SQL Server	v6.x o superior	Controlador JDBC de Microsoft para SQL Server	No compatible
MySQL	v5.1 o superior	Conectores MySQL	No compatible
Oracle	v11.2 o superior	Descargas de Oracle JDBC	No compatible
PostgreSQL	v4.2.x o superior	Controlador JDBC de PostgreSQL	No compatible
Amazon Redshift	v4.1 o superior	Conexión a Amazon Redshift con JDBC	compatible

Producto	Versión de compatible	Instrucciones y descargas del controlador	Se admiten consultas SQL
Snowflake	Para ver su versión de Snowflake, utilice CURRENT_VERSION tal y como se describe en la documentación de Snowflake.	Para conectarse a Snowflake, necesita lo siguiente: • Controlador JDBC Snowflake • Conector Snowflake para Spark	compatible

Para conectarse a bases de datos o almacenes de datos que requieren una versión del controlador diferente a la compatible de DataBrew forma nativa, puede proporcionar el controlador JDBC de su elección. El controlador debe ser compatible con JDK 8 o Java 8. Para obtener instrucciones sobre cómo encontrar la versión más reciente del controlador para su base de datos, consulteUso de controladores con AWS Glue DataBrew.

Conectarse a los datos de un archivo de texto con DataBrew

Puede configurar las siguientes opciones de formato para los archivos de entrada DataBrew compatibles:

- Comma-separated archivos de valores (CSV)
  - Delimitadores

El delimitador predeterminado es una coma para los archivos.csv. Si el archivo usa un delimitador diferente, elija el delimitador para el delimitador CSV en la sección Configuraciones adicionales al crear el conjunto de datos. Los siguientes delimitadores son compatibles con los archivos.csv:

- Coma (,)
- Dos puntos (:)
- Semi-colon (;)
- Barra vertical (|)
- Tabulador (\t)
- Caret (^)
- Barra invertida (\)
- Space

- Valores del encabezado de columna

  El archivo CSV puede incluir una fila de encabezado como primera fila del archivo. Si no es así, DataBrew crea una fila de encabezado para ti.

  - Si el archivo CSV incluye una fila de encabezado, selecciona Tratar la primera fila como encabezado. Si lo hace, se considerará que la primera fila del archivo CSV contiene los valores del encabezado de la columna.
  - Si el archivo CSV no incluye una fila de encabezado, selecciona Añadir encabezado predeterminado. Si lo hace, DataBrew crea una fila de encabezado para el archivo y no considera que la primera fila de datos contenga valores de encabezado. Los encabezados que DataBrew crea constan de un guión bajo y un número para cada columna del archivo, en el formato `Column_1Column_2`,`Column_3`,, etc.

- Archivos JSON

  DataBrew admite dos formatos para archivos JSON: líneas JSON y documentos JSON. Los archivos de líneas JSON contienen una fila por línea. En los archivos de documentos JSON, todas las filas están contenidas en una única estructura JSON o en una matriz. Puede especificar el tipo de archivo JSON en la sección Configuraciones adicionales al crear un conjunto de datos JSON. El formato predeterminado es JSON Lines.

- Archivos de Excel

  Lo siguiente se aplica a las hojas de Excel en DataBrew:

- Carga de hojas de Excel

  De forma predeterminada, DataBrew carga la primera hoja del archivo de Excel. Sin embargo, puede especificar un número de hoja o un nombre de hoja diferente en la sección Configuraciones adicionales al crear un conjunto de datos de Excel.

- Valores del encabezado de columna

  Sus hojas de Excel pueden incluir una fila de encabezado como primera fila del archivo, pero si no lo hacen, DataBrew crearán una fila de encabezado automáticamente.

  - Si sus hojas de Excel incluyen una fila de encabezado, elija Tratar la primera fila como encabezado. Si lo hace, se considerará que la primera fila de las hojas de Excel contiene los valores del encabezado de las columnas.
  - Si el archivo de Excel no incluye una fila de encabezado, selecciona Añadir encabezado predeterminado. De este modo, especificas que se DataBrew debe crear una fila de encabezado para el archivo y no tratar la primera fila de datos como si contuviera valores de encabezado. Los encabezados que se DataBrew crean constan de un guión bajo y un número para cada columna del archivo, en el formato `Column_1Column_2`,`Column_3`,, etc.

Conexión de datos de varios archivos en Amazon S3

Con la DataBrew consola, puede navegar por los depósitos y carpetas de Amazon S3 y elegir un archivo para su conjunto de datos. Sin embargo, no es necesario que un conjunto de datos esté limitado a un archivo.

Supongamos que tiene un bucket de S3 con un nombre `my-databrew-bucket` que contiene una carpeta denominada`databrew-input`. En esa carpeta, supongamos que tiene varios archivos JSON, todos con el mismo formato y extensión de `.json` archivo. En la consola, puede especificar una URL de origen de`s3://my-databrew-bucket/databrew-input/`. A continuación, en la DataBrew consola, puede elegir esta carpeta. El conjunto de datos se compone de todos los archivos JSON de esa carpeta.

DataBrew puede procesar todos los archivos de una carpeta S3, pero solo si se cumplen las siguientes condiciones:

- Todos los archivos de la carpeta tienen el mismo formato.
- Todos los archivos de la carpeta tienen la misma extensión.

Para obtener más información sobre los formatos y extensiones de archivo compatibles, consulteDataBrew input formats.

Esquemas cuando se utilizan varios archivos como conjunto de datos

Cuando se utilizan varios archivos como DataBrew conjunto de datos, los esquemas deben ser los mismos en todos los archivos. De lo contrario, el espacio de trabajo del proyecto intentará elegir automáticamente uno de los esquemas de los múltiples archivos e intentará ajustar el resto de los archivos del conjunto de datos a ese esquema. Este comportamiento hace que la vista que se muestra en Project Workspace sea irregular y, en consecuencia, el resultado del trabajo también lo será.

Si los archivos deben tener esquemas diferentes, debe crear varios conjuntos de datos y perfilarlos por separado.

Uso de rutas parametrizadas para Amazon S3

En algunos casos, es posible que desee crear un conjunto de datos con archivos que sigan una determinada convención de nomenclatura o un conjunto de datos que pueda abarcar varias carpetas de Amazon S3. O puede que desee reutilizar el mismo conjunto de datos para datos con una estructura idéntica que se generan periódicamente en una ubicación de S3 con una ruta que depende de determinados parámetros. Un ejemplo es una ruta cuyo nombre corresponde a la fecha de producción de los datos.

DataBrew admite este enfoque con rutas S3 parametrizadas. Una ruta parametrizada es una URL de Amazon S3 que contiene expresiones regulares o parámetros de ruta personalizados, o ambos.

Definir un conjunto de datos con una ruta S3 mediante expresiones regulares

Las expresiones regulares de la ruta pueden resultar útiles para hacer coincidir varios archivos de una o más carpetas y, al mismo tiempo, filtrar los archivos no relacionados de esas carpetas.

He aquí un par de ejemplos:

- Defina un conjunto de datos que incluya todos los archivos JSON de una carpeta cuyo nombre comience por`invoice`.
- Defina un conjunto de datos que incluya todos los archivos de las carpetas con `2020` sus nombres.

Puede implementar este tipo de enfoque mediante el uso de expresiones regulares en la ruta S3 de un conjunto de datos. Estas expresiones regulares pueden reemplazar cualquier subcadena de la clave de la URL de S3 (pero no el nombre del bucket).

Como ejemplo de una clave en una URL de S3, consulta lo siguiente. Aquí `my-bucket` está el nombre del bucket, US East (Ohio) es la AWS región y `puppy.png` es el nombre de la clave.

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

En una ruta S3 parametrizada, todos los caracteres entre dos corchetes angulares (<y>) se tratan como expresiones regulares. A continuación se muestran dos ejemplos:

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json`hace coincidir todos los archivos con nombres `data.json` incluidos en todas las subcarpetas `databrew-input` cuyos nombres comiencen por. `invoice`
- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/`hace coincidir todos los archivos de las carpetas con `2020` sus nombres.

En estos ejemplos, `.*` coincide con cero o más caracteres.

> **Note**
>
> Solo puedes usar expresiones regulares en la parte clave de la ruta S3, la parte que va después del nombre del bucket. Por lo tanto, `s3://my-databrew-bucket/<.*>-input/` es válido, pero `s3://my-<.*>-bucket/<.*>-input/` no lo es.

Le recomendamos que pruebe sus expresiones regulares para asegurarse de que solo coincidan con las URL de S3 que desee y no con las que no desee.

Estos son algunos otros ejemplos de expresiones regulares:

- `<\d{2}>`coincide con una cadena que consta exactamente de dos dígitos consecutivos, por ejemplo `07` o`03`, pero no`1a2`.
- `<[a-z]+.*>`coincide con una cadena que comienza con una o más letras latinas minúsculas y tiene cero o más caracteres después de ella. Un ejemplo es`a3`, o `abc/defa-z`, pero no. `A2`
- `<[^/]+>`coincide con una cadena que contiene cualquier carácter excepto una barra (/). En una URL de S3, las barras se utilizan para separar las carpetas de la ruta.

- `<.*=.*>`coincide con una cadena que contiene un signo igual (=), por ejemplo`month=02`, o `abc/day=2=10`, pero no. `test`
- `<\d.*\d>`coincide con una cadena que comienza y termina con un dígito y puede tener cualquier otro carácter entre los dígitos, por ejemplo `1abc201-02-03`, o`2020/Jul/21`, pero no`123a`.

Definir un conjunto de datos con una ruta S3 mediante parámetros personalizados

Definir un conjunto de datos parametrizado mediante parámetros personalizados ofrece ventajas en comparación con el uso de expresiones regulares cuando es posible que desee proporcionar parámetros para una ubicación de S3:

- Puede obtener los mismos resultados que con una expresión regular, sin necesidad de conocer la sintaxis de las expresiones regulares. Puede definir los parámetros con términos conocidos, como «empieza por» y «contiene».
- Al definir un conjunto de datos dinámico con los parámetros de la ruta, puede incluir un intervalo de tiempo en la definición, como «el mes pasado» o «las últimas 24 horas». De esta forma, la definición de tu conjunto de datos se utilizará más adelante con los nuevos datos entrantes.

Estos son algunos ejemplos de cuándo es posible que desee utilizar conjuntos de datos dinámicos:

- Para conectar varios archivos que están particionados por la fecha de la última actualización u otros atributos significativos en un único conjunto de datos. A continuación, puede capturar estos atributos de partición como columnas adicionales en un conjunto de datos.
- Restringir los archivos de un conjunto de datos a ubicaciones de S3 que cumplan determinadas condiciones. Por ejemplo, supongamos que su ruta de S3 contiene carpetas basadas en fechas como`folder/2021/04/01/`. En este caso, puedes parametrizar la fecha y restringirla a un intervalo determinado, como «entre el 1 de marzo de 2021 y el 1 de abril de 2021» o «La semana pasada».

Para definir una ruta mediante parámetros, defina los parámetros y agréguelos a su ruta con el siguiente formato:

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

> **Note**
>
> Al igual que ocurre con las expresiones regulares en una ruta de S3, solo puede usar parámetros en la parte clave de la ruta, es decir, la parte que va después del nombre del bucket.

Se requieren dos campos en la definición, el nombre y el tipo de un parámetro. El tipo puede ser Cadena, Número o Fecha. Los parámetros de tipo Fecha deben tener una definición del formato de fecha para poder DataBrew interpretar y comparar correctamente los valores de fecha. Si lo desea, puede definir las condiciones de coincidencia para un parámetro. También puede optar por añadir los valores coincidentes de un parámetro como una columna a su conjunto de datos cuando lo cargue un DataBrew trabajo o una sesión interactiva.

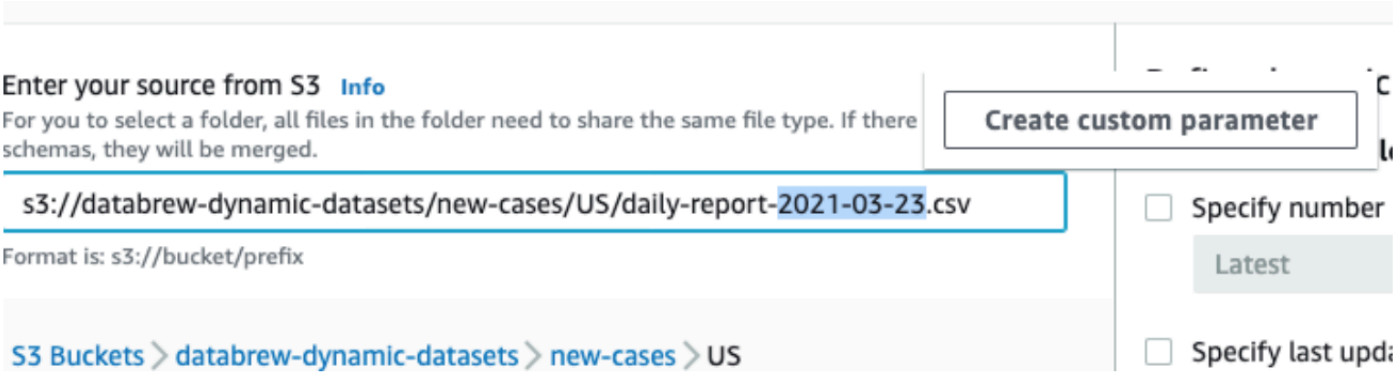
Ejemplo

Consideremos un ejemplo de definición de un conjunto de datos dinámico mediante parámetros de la DataBrew consola. En este ejemplo, supongamos que los datos de entrada se escriben normalmente en un bucket de S3 utilizando ubicaciones como las siguientes:

- `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`

Aquí hay dos partes dinámicas: un código de país, como EE. UU., y una fecha en el nombre del archivo, como el 30 de marzo de 2021. Aquí puede aplicar la misma receta de limpieza para todos los archivos. Supongamos que desea realizar su trabajo de limpieza a diario. A continuación se muestra cómo se puede definir una ruta parametrizada para este escenario:

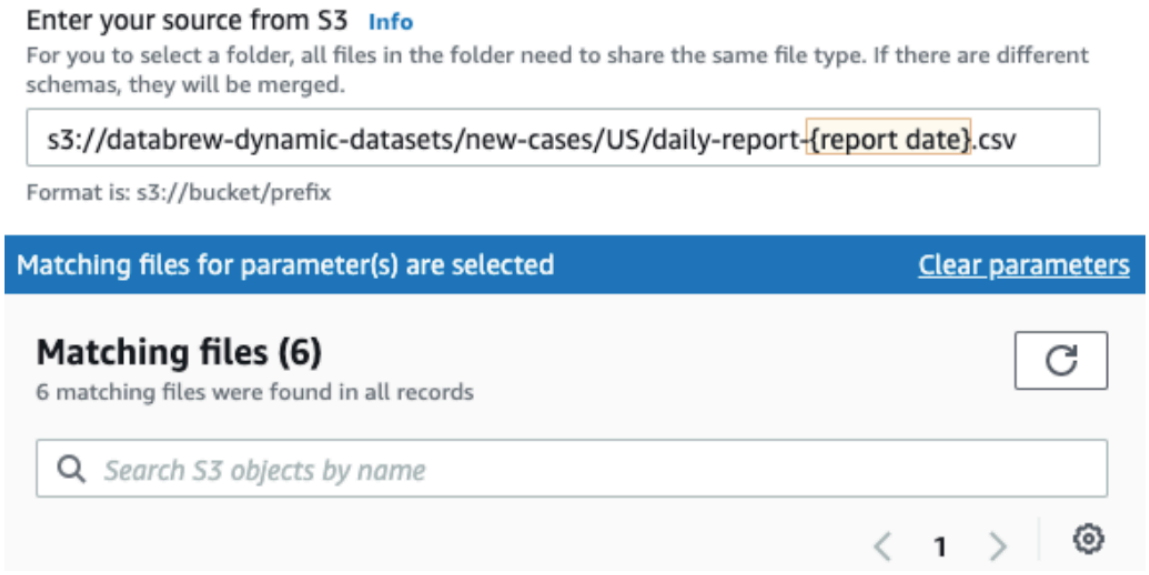
1. Navegue hasta un archivo específico.
2. A continuación, seleccione una parte variable, como una fecha, y sustitúyala por un parámetro. En este caso, sustituya una fecha.



3. Abra el menú contextual (haga clic con el botón derecho) de Crear un parámetro personalizado y defina sus propiedades:
   - Nombre: fecha del informe
   - Tipo: fecha
   - Formato de fecha: aaaa- MM-dd (seleccionado entre los formatos predefinidos)
   - Condiciones (intervalo de tiempo): últimas 24 horas
   - Añadir como columna: verdadero (marcado)

   Mantenga los demás campos con sus valores predeterminados.
4. Seleccione Crear.

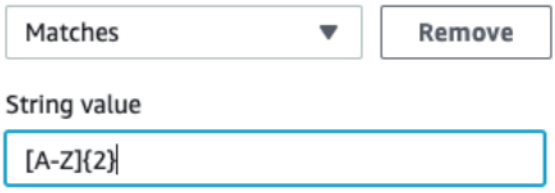
Después de hacerlo, verá la ruta actualizada, como en la siguiente captura de pantalla.



Ahora puede hacer lo mismo con el código de país y parametrizarlo de la siguiente manera:

- Nombre: código de país
- Tipo: cadena
- Añadir como columna: verdadero (marcado)

No es necesario especificar condiciones si todos los valores son relevantes. En la `new-cases` carpeta, por ejemplo, solo tenemos subcarpetas con códigos de país, por lo que no se necesitan condiciones. Si tuvieras que excluir otras carpetas, podrías usar la siguiente condición.



Este enfoque limita las subcarpetas de los casos nuevos para que contengan dos caracteres latinos en mayúscula.

Tras esta parametrización, solo tendrá los archivos coincidentes en nuestro conjunto de datos y podrá elegir Crear conjunto de datos.

> **Note**
>
> Cuando utilizas rangos de tiempo relativos en condiciones, los rangos de tiempo se evalúan cuando se carga el conjunto de datos. Esto es cierto tanto si se trata de intervalos de tiempo predefinidos, como «Últimas 24 horas», como de intervalos de tiempo personalizados, como «Hace 5 días». Este enfoque de evaluación se aplica tanto si el conjunto de datos se carga durante la inicialización de una sesión interactiva como si se inicia un trabajo.

Después de elegir Crear conjunto de datos, el conjunto de datos dinámico estará listo para usarse. Por ejemplo, puede usarlo primero para crear un proyecto y definir una receta de limpieza mediante una DataBrew sesión interactiva. Luego, puede crear un trabajo que esté programado para ejecutarse a diario. Este trabajo podría aplicar la receta de limpieza a los archivos del conjunto de datos que cumplen las condiciones de sus parámetros en el momento en que se inicia el trabajo.

Condiciones compatibles para los conjuntos de datos dinámicos

Puede usar condiciones para filtrar los archivos S3 coincidentes mediante parámetros o el atributo de fecha de la última modificación.

A continuación, encontrará listas de condiciones admitidas para cada tipo de parámetro.

Condiciones utilizadas con los parámetros de cadena

Nombre en el DataBrew SDK	Sinónimos del SDK	Nombre en la DataBrew consola	Description (Descripción)
es	eq, ==	Es exactamente	El valor del parámetro es el mismo que el valor que se proporcionó en la condición.
no lo es	no es igual,! =	Is not	El valor del parámetro no es el mismo que el valor que se proporcionó en la condición.
contains		Contiene	El valor de cadena del parámetro contiene el valor que se proporcionó en la condición.
no contiene		No contiene	El valor de cadena del parámetro no contiene el valor que se proporcionó en la condición.
comienza_con		Empieza por	El valor de cadena del parámetro comienza con el valor que se proporcionó en la condición.
no comienza con		No comienza con	El valor de cadena del parámetro no comienza por el valor que se proporcionó en la condición.

Nombre en el DataBrew SDK	Sinónimos del SDK	Nombre en la DataBrew consola	Description (Descripción)
ends_with		Acaba con	El valor de cadena del parámetro termina con el valor que se proporcionó en la condición.
no termina con_con		No termina con	El valor de cadena del parámetro no termina con el valor que se proporcionó en la condición.
matches		Coincide	El valor del parámetro coincide con la expresión regular proporcionada en la condición.
no coincide		No coincide	El valor del parámetro no coincide con la expresión regular proporcionada en la condición.

Note

Todas las condiciones de los parámetros de cadena utilizan una comparación que distingue entre mayúsculas y minúsculas. Si no está seguro de las mayúsculas y minúsculas utilizadas en una ruta de S3, puede utilizar la condición «coincide» con un valor de expresión regular que comience por. `(?i)` De este modo, se obtiene una comparación que no distingue entre mayúsculas y minúsculas.

Por ejemplo, supongamos que quiere que el parámetro de cadena comience por`abc`, pero `Abc` también `ABC` es posible. En este caso, puede usar la condición «coincide» `(?i)^abc` como valor de condición.

Condiciones utilizadas con los parámetros numéricos

Nombre en el DataBrew SDK	Sinónimos del SDK	Nombre en la DataBrew consola	Description (Descripción)
es	eq, ==	Es exactamente	El valor del parámetro es el mismo que el valor que se proporcionó en la condición.
no lo es	no es igual,! =	Is not	El valor del parámetro no es el mismo que el valor que se proporcionó en la condición.
meno_que	mucho, <	Menor que	El valor numérico del parámetro es inferior al valor que se proporcionó en la condición.
meno_que_igual	LTE, <=	Menor o igual que	El valor numérico del parámetro es menor o igual al valor que se proporcionó en la condición.
mayor_que	gt, >	Mayor que	El valor numérico del parámetro es mayor que el valor que se proporcionó en la condición.

Nombre en el DataBrew SDK	Sinónimos del SDK	Nombre en la DataBrew consola	Description (Descripción)
mayor_que_igual	obtener, >=	Mayor o igual que	El valor numérico del parámetro es mayor o igual al valor que se proporcionó en la condición.

Condiciones utilizadas con los parámetros de fecha

Nombre en el DataBrew SDK	Nombre en la DataBrew consola	Formato de valor de condición (SDK)	Description (Descripción)
después	Inicio	Formato de fecha ISO 8601 similar `2021-03-30T01:00:00Z` o `2021-03-30T01:00-07:00`	El valor del parámetro de fecha es posterior a la fecha proporcionada en la condición.
antes	Final	Formato de fecha ISO 8601 similar `2021-03-30T01:00:00Z` o `2021-03-30T01:00-07:00`	El valor del parámetro de fecha es anterior a la fecha proporcionada en la condición.
relative_después	Inicio (relativo)	Número positivo o negativo de unidades de tiempo, como `-48h` o `+7d`.	El valor del parámetro de fecha es posterior a la fecha relativa proporcionada en la condición. Las fechas relativas se evalúan cuando se carga el conjunto

Nombre en el DataBrew SDK	Nombre en la DataBrew consola	Formato de valor de condición (SDK)	Description (Descripción)
			de datos, ya sea cuando se inicializa una sesión interactiva o cuando se inicia un trabajo asociado. Este es el momento que se denomina «ahora» en los ejemplos.
relative_antes	Fin (relativo)	Número positivo o negativo de unidades de tiempo, como `-48h` o `+7d`.	El valor del parámetro de fecha es anterior a la fecha relativa proporcionada en la condición. Las fechas relativas se evalúan cuando se carga el conjunto de datos, ya sea cuando se inicializa una sesión interactiva o cuando se inicia un trabajo asociado. Este es el momento que se denomina «ahora» en los ejemplos.

Si usa el SDK, proporcione fechas relativas en el siguiente formato:`±{number_of_time_units}{time_unit}`. Puedes usar estas unidades de tiempo:

- -1h (hace 1 hora)
- +2d (dentro de 2 días)
- -120 m (hace 120 minutos)

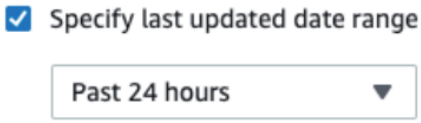
- 5000 s (dentro de 5000 segundos)
- -3w (hace 3 semanas)
- +4M (dentro de 4 meses)
- -1y (hace 1 año)

Las fechas relativas se evalúan cuando se carga el conjunto de datos, ya sea cuando se inicializa una sesión interactiva o cuando se inicia un trabajo asociado. Este es el momento que se denomina «ahora» en los ejemplos anteriores.

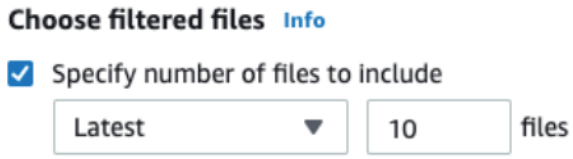
Configurar los ajustes de los conjuntos de datos dinámicos

Además de proporcionar una ruta S3 parametrizada, puede configurar otros ajustes para conjuntos de datos con varios archivos. Estas configuraciones filtran los archivos S3 por su fecha de última modificación y limitan el número de archivos.

De forma similar a la configuración de un parámetro de fecha en una ruta, puede definir un intervalo de tiempo en el que se actualizaron los archivos coincidentes e incluir solo esos archivos en su conjunto de datos. Puede definir estos rangos utilizando fechas absolutas, como el «30 de marzo de 2021», o rangos relativos, como «Últimas 24 horas».



Para limitar el número de archivos coincidentes, seleccione un número de archivos superior a 0 e indique si desea los archivos coincidentes más recientes o los más antiguos.



Tipos de datos

Los datos de cada columna del conjunto de datos se convierten a uno de los siguientes tipos de datos:

- byte: números enteros con signo de 1 byte. El rango de números va de -128 a 127.
- corto: números enteros con signo de 2 bytes. El rango de números va de -32768 a 32767.

- entero: números enteros con signo de 4 bytes. El rango de números va de -2147483648 a 2147483647.
- long: números enteros con signo de 8 bytes. El rango de números va de -9223372036854775808 a 9223372036854775807.
- float: números de coma flotante de precisión simple de 4 bytes.
- doble: números de coma flotante de precisión doble de 8 bytes.
- decimal: números decimales con signo con hasta 38 dígitos en total y 18 dígitos después de la coma decimal.
- cadena: valores de cadena de caracteres.
- booleano: el tipo booleano tiene uno de dos valores posibles: `verdadero` y `falso` o `sí` y `no`.
- marca de tiempo: valores que comprenden los campos año, mes, día, hora, minuto y segundo.
- fecha: valores que comprenden los campos año, mes y día.

Tipos de datos avanzados

Los tipos de datos avanzados son tipos de datos que se DataBrew detectan dentro de una columna de cadena en un proyecto y, por lo tanto, no forman parte de un conjunto de datos. Para obtener información sobre los tipos de datos avanzados, consulte Tipos de datos avanzados.

Tipos de datos avanzados

Los tipos de datos avanzados son tipos de datos que se DataBrew detectan dentro de una columna de cadena de un proyecto mediante la coincidencia de patrones. Al hacer clic en una columna de cadena, la columna se marca como el tipo de datos avanzado correspondiente si el 50% o más de los valores de la columna cumplen los criterios de ese tipo de datos.

Los tipos de datos que DataBrew se pueden detectar son:

- Date/timestamp
- SSN
- Número de teléfono
- Correo electrónico
- Tarjeta de crédito
- Gender

- Dirección IP
- URL
- Código postal
- País
- Currency (Divisa)
- Estado
- Ciudad

Puede utilizar las siguientes transformaciones para trabajar con tipos de datos avanzados:

- GET_ADVANCED_DATATYPE: dada una columna de cadena, identifica el tipo de datos avanzado de la columna, si lo hay.
- EXTRACT_ADVANCED_DATATYPE_DETAILS: extrae los detalles de un tipo de datos avanzado.
- FILTRO_DE_TIPO_DATOS AVANZADO: filtra una columna de origen actual en función de la detección avanzada de tipos de datos.
- ADVANCED_DATATYPE_FLAG: crea una nueva columna indicadora basada en los valores de la columna de origen actual.

Validación de la calidad de los datos en AWS Glue DataBrew

Para garantizar la calidad de sus conjuntos de datos, puede definir una lista de reglas de calidad de los datos en un conjunto de reglas. Un conjunto de reglas es un conjunto de reglas que comparan diferentes métricas de datos con los valores esperados. Si no se cumple alguno de los criterios de una regla, el conjunto de reglas en su conjunto no pasa la validación. A continuación, puede inspeccionar los resultados individuales de cada regla. En el caso de cualquier regla que provoque un error de validación, puede realizar las correcciones necesarias y volver a validarla.

Entre los ejemplos de reglas se incluyen los siguientes:

- El valor de la columna `"APY"` está entre 0 y 100
- El número de valores faltantes en la columna `group_name` no supera el 5%

Puede definir cada regla para una columna individual o aplicarla de forma independiente a varias columnas seleccionadas, por ejemplo:

- El valor máximo no supera los 100 para las columnas`"rate"`,`"pay"`,`"increase"`.

Una regla puede consistir en varias comprobaciones sencillas. Puede definir si todas deben ser verdaderas o alguna, por ejemplo:

- El valor de la columna `"ProductId"` debe empezar por `"asin-"` Y la longitud del valor de la columna `"ProductId"` debe ser 32.

Puede comprobar las reglas con valores agregados (por ejemplo `maxmin`, o si solo `number of duplicate values` se compara un valor) o con valores no agregados en cada fila de una columna. En este último caso, también puede definir un umbral «válido», como`value in columnA > value in columnB for at least 95% of rows`.

Al igual que con la información de perfil, puede definir reglas de calidad de datos a nivel de columna solo para columnas de tipos simples, como cadenas y números. No puede definir reglas de calidad de datos para columnas de tipos complejos, como matrices o estructuras. Para obtener más información sobre cómo trabajar con información de perfil, consulteCrear y trabajar con AWS Glue DataBrew trabajos de perfil.

Validación de las reglas de calidad de los datos

Una vez definido un conjunto de reglas, puede añadirlo a un trabajo de perfil para su validación. Puede definir más de un conjunto de reglas para un conjunto de datos.

Por ejemplo, un conjunto de reglas puede contener reglas con criterios mínimamente aceptables. Un error en la validación de ese conjunto de reglas podría significar que los datos no son aceptables para su uso posterior. Un ejemplo es la falta de valores en las columnas clave de un conjunto de datos utilizado para el entrenamiento del aprendizaje automático. Puedes usar un segundo conjunto de reglas con reglas más estrictas para verificar si el conjunto de datos tiene una calidad tan buena que no sea necesario limpiarlo.

Puede aplicar uno o más conjuntos de reglas definidos para un conjunto de datos determinado en una configuración de trabajo de perfil. Cuando se ejecuta el trabajo de perfil, genera un informe de validación además del perfil de datos. El informe de validación está disponible en la misma ubicación que los datos de su perfil. Al igual que con la información del perfil, puede explorar los resultados en la DataBrew consola. En la vista de detalles del conjunto de datos, seleccione la pestaña Calidad de los datos para ver los resultados. Para obtener más información sobre cómo trabajar con la información del perfil, consulteCrear y trabajar con AWS Glue DataBrew trabajos de perfil.

Actuar sobre los resultados de la validación

Cuando se completa un trabajo de DataBrew perfil, DataBrew envía un CloudWatch evento de Amazon con los detalles de la ejecución de ese trabajo. Si también configuró su trabajo para validar las reglas de calidad de los datos, DataBrew envía un evento para cada conjunto de reglas validado. El evento contiene su resultado (`SUCCEEDEDFAILED`, o`ERROR`) y un enlace al informe detallado de validación de la calidad de los datos. A continuación, puede automatizar cualquier acción posterior invocando la acción siguiente en función del estado de la validación. Para obtener más información sobre cómo conectar eventos con acciones segmentadas, como notificaciones de Amazon SNS, invocaciones de AWS Lambda funciones y otras, consulte Cómo empezar a usar Amazon. EventBridge

A continuación se muestra un ejemplo de un evento de resultado de DataBrew validación:

```
{
  "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"
  }
}
```

Puede utilizar los atributos de los eventos`detail-type`, `source` como las propiedades anidadas del `detail` atributo, para crear patrones de eventos en Amazon Eventbridge. Por ejemplo, un patrón de eventos que coincida con todas las validaciones fallidas de cualquier DataBrew trabajo tendría el siguiente aspecto:

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

Para ver un ejemplo de cómo crear un conjunto de reglas y validar sus reglas, consulte. Crear un conjunto de reglas con reglas de calidad de datos Para obtener más información sobre cómo trabajar con CloudWatch eventos en, consulte DataBrew Automatizar DataBrew con eventos CloudWatch

Crear un conjunto de reglas con reglas de calidad de datos

En el siguiente procedimiento, encontrará un ejemplo de cómo crear un conjunto de reglas y aplicarlo a un conjunto de datos. Un conjunto de reglas es un conjunto de reglas que comparan diferentes métricas de datos con los valores esperados. A continuación, puede utilizar este conjunto de reglas en un trabajo de perfil para validar las reglas de calidad de los datos que incluye.

Para crear un conjunto de reglas de ejemplo con reglas de calidad de datos

1. Inicie sesión en Consola de administración de AWS y abra la DataBrew consola en. https://console.aws.amazon.com/databrew/
2. Elija DQ RULES en el panel de navegación y, a continuación, elija Crear conjunto de reglas de calidad de datos.
3. Introduzca un nombre para el conjunto de reglas. Si lo desea, introduzca una descripción para el conjunto de reglas.
4. En Conjunto de datos asociado, elige un conjunto de datos para asociarlo al conjunto de reglas.

   Después de seleccionar un conjunto de datos, puede ver el panel de vista previa del conjunto de datos a la derecha.
5. Utilice la vista previa del panel de vista previa del conjunto de datos para explorar los valores y el esquema del conjunto de datos a medida que determina las reglas de calidad de los datos que va a crear. La vista previa puede proporcionarle información sobre los posibles problemas que pueda tener con los datos.

   Algunas fuentes de datos, como las bases de datos, no admiten la vista previa de los datos. En ese caso, puede ejecutar un trabajo de perfil sin validar primero las reglas de calidad de los datos. A continuación, puede obtener información sobre el esquema de datos y la distribución de valores mediante el perfil de datos.
6. Consulte la pestaña Recomendaciones, en la que se muestran algunas sugerencias de reglas que puede utilizar al crear su conjunto de reglas. Puede seleccionar todas las recomendaciones, algunas o ninguna.

   Tras seleccionar las recomendaciones pertinentes, elija Añadir al conjunto de reglas.

   Esto añadirá reglas a tu conjunto de reglas. Inspeccione y modifique los parámetros si es necesario. Tenga en cuenta que en las reglas de calidad de los datos solo se pueden usar columnas de tipos simples, como cadenas, números y valores booleanos.
7. Seleccione Añadir otra regla para añadir una regla que no esté incluida en las recomendaciones. Puede cambiar los nombres de las reglas para facilitar la interpretación posterior de los resultados de la validación.
8. Utilice el ámbito de control de calidad de los datos para elegir si se seleccionarán columnas individuales por cada comprobación de esta regla o si se deben aplicar a un grupo de columnas que seleccione. Por ejemplo, si su conjunto de datos tiene varias columnas numéricas que

deben tener valores entre 0 y 100, puede definir la regla una vez y seleccionar todas estas columnas para que se comprueben según esta regla.

9. Si la regla va a tener más de una comprobación, en el menú desplegable Criterios de cumplimiento de la regla, elija si se deben cumplir todas las comprobaciones o cuáles cumplen los criterios.
10. Seleccione la comprobación que se realizará para verificar esta regla en el menú desplegable de control de calidad de los datos. Para obtener más información sobre las comprobaciones disponibles, consulteCheques disponibles.
11. Si eligió Comprobación individual para cada columna del ámbito de control de calidad de los datos, elija una columna. Seleccione o escriba el nombre de la columna para esta comprobación.
12. Seleccione los parámetros en función de la comprobación. Algunas condiciones solo aceptan los valores personalizados proporcionados y otras también admiten la referencia a otra columna.
13. Si selecciona los valores de columna, como la condición Contiene para los valores de cadena, puede especificar el umbral de «superación». Por ejemplo, si desea que al menos el 95 por ciento de los valores cumplan la condición, debe elegir Mayor que igual como condición de umbral, introducir 95 como umbral y dejar «% (porcentaje) de filas» en el siguiente menú desplegable de la sección Umbral. O si no quieres más de 10 filas en las que falte un valor, la condición es verdadera, puedes seleccionar Menos que igual como condición, introducir 10 como Umbral y elegir filas en el siguiente menú desplegable. Ten en cuenta que es posible que obtengas resultados diferentes si utilizas muestras de diferentes tamaños durante la validación.
14. Agrega más reglas si es necesario.
15. Selecciona Crear conjunto de reglas.

Crear un trabajo de perfil mediante un conjunto de reglas

Tras crear un conjunto de reglas como se ha descrito anteriormente, accederá a la página de normas de calidad de los datos, en la que se muestran todos los conjuntos de reglas de su cuenta.

Para crear un trabajo de perfil que incluya un conjunto de reglas

1. Elija el nombre del conjunto de reglas que creó anteriormente para ver sus detalles.
2. Elija Crear un trabajo de perfil con un conjunto de reglas.

   El nombre del trabajo se rellena automáticamente, pero puede cambiarlo según sea necesario.

3. Para el ejemplo de ejecución de Job, puede elegir ejecutar todo el conjunto de datos o un número limitado de filas.

   Si opta por utilizar un tamaño de muestra limitado, tenga en cuenta que, según ciertas reglas, los resultados pueden diferir en comparación con el conjunto de datos completo.

4. En la configuración de salida del trabajo, elija una ubicación S3 para el resultado del trabajo. Elija cualquier carpeta de un bucket de Amazon S3 con nombre al que tenga acceso. Si introduce un nombre de carpeta para este depósito que no existe, se crea esta carpeta.

   Al completar correctamente el trabajo de perfil, esta carpeta contendrá los perfiles de los datos y el informe de validación de las reglas de calidad de los datos en formato JSON.

5. En Reglas de calidad de datos, tenga en cuenta que su conjunto de reglas aparece en el nombre del conjunto de reglas de calidad de datos.

6. En Permisos, seleccione o cree un rol para conceder DataBrew acceso a leer desde la ubicación de entrada de Amazon S3 y escribir en la ubicación de salida del trabajo. Si aún no tiene un rol preparado, seleccione Crear un nuevo rol de IAM.

7. Modifique cualquier otra configuración opcional como se describe enCrear y trabajar con AWS Glue DataBrew trabajos de perfil, si es necesario.

8. Elija Crear y ejecutar trabajo.

Inspeccionar los resultados de la validación y actualizar las reglas de calidad de los datos

Una vez finalizado el trabajo de perfil, podrá ver los resultados de la validación de las reglas de calidad de los datos y, si es necesario, actualizar las reglas.

Para ver los datos de validación de sus reglas de calidad de datos

1. En la DataBrew consola, selecciona Ver perfil de datos. Al hacer esto, se muestra la pestaña de descripción general del perfil de datos de su conjunto de datos.

2. Elija la pestaña Reglas de calidad de los datos. En esta pestaña, puede ver los resultados de todas las reglas de calidad de los datos.

3. Seleccione una regla individual para obtener más detalles sobre esa regla.

Para cualquier regla que no se haya validado, puede realizar las correcciones necesarias.

Para actualizar las normas de calidad de los datos

1. En el panel de navegación, elija DQ RULES.
2. En Nombre del conjunto de reglas de calidad de datos, elija el conjunto de datos que contiene las reglas que planea editar.
3. Elija la regla que desee cambiar y, a continuación, elija Editar.
4. Realice las correcciones necesarias y, a continuación, seleccione Actualizar conjunto de reglas.
5. Vuelva a ejecutar el trabajo. Repita este proceso hasta que se aprueben todas las validaciones.

Cheques disponibles

En la siguiente tabla se enumeran las referencias de todas las condiciones disponibles que se pueden utilizar en las reglas. Tenga en cuenta que las condiciones agregadas no se pueden combinar con condiciones no agregadas en la misma regla.

> **Note**
>
> Para los usuarios del SDK, para aplicar la misma regla a varias columnas, utilice el ColumnSelectorsatributo de una regla y especifique las columnas validadas mediante sus nombres o una expresión regular. En este caso, debes usar implícita CheckExpression. Por ejemplo, `"> :val"` para comparar los valores de cada una de las columnas seleccionadas con el valor proporcionado. DataBrew utiliza una sintaxis implícita para la definición FilterExpressionen conjuntos de datos dinámicos. Si quiere especificar columnas para cada comprobación de forma individual, no defina el ColumnSelectorsatributo. En su lugar, proporcione una expresión explícita. Por ejemplo, `":col > :val"` como CheckExpressionen una regla.

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK
Condiciones de conjuntos de datos agregados	Número de filas		Comparación numérica con el valor personalizado	`"CheckExpression": "AGG(ROWS_COUNT)`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK
				`> :val", "SubstitutionMap": {":val", "10000"}`
	Número de columnas		Comparación numérica con el valor personalizado	`"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Filas duplicadas		Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

Condiciones de estadísticas de columnas agregadas	Valores faltantes		Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores duplicados		Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores válidos		Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
				
				o
				
				`"CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

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

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores únicos		Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores atípicos	Z-score umbral	Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Estadísticas de distribución de valores	Nombre de las estadísticas (consulte la siguiente tabla)	Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Consulte la siguiente tabla para ver los `STAT_NAME`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK
				valores posibles

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Estadísticas numéricas	Nombre de la estadística (consulte la siguiente tabla)	Comparación numérica con un valor personalizado	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				o
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Consulte la siguiente tabla para ver los `STAT_NAME`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK
				valores posibles
No agregado (acepta el umbral)	El valor es exacto		Comparación exacta con una lista de valores	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	El valor no es exacto		El valor no debe coincidir exactamente con ningún valor de una lista	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores de cadena		Comparación de cadenas con un valor personalizado u otra columna de cadena	`"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"}`
				
				o
				
				`"CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Valores numéricos		Comparación numérica con un valor personalizado u otra columna numérica	`"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"}`
				
				o
				
				`"CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}`

Tipo de condición	Verificación de la calidad de los datos	Parámetros adicionales	Tipo de comparación	Ejemplo de sintaxis del SDK

	Longitud de la cadena de valores		Comparación numérica con un valor personalizado u otra columna numérica	`"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"}`
				
				o
				
				`"CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}`

Comparaciones numéricas

DataBrew admite las siguientes operaciones para la comparación numérica: Es igual a (= =), No es igual a (! =), Menor que (<), Menor que igual a (< =), Mayor que (>), Mayor que igual (> =) y Está entre (is_entre:val1 y:val2).

Comparaciones de cadenas

Se admiten las siguientes comparaciones de cadenas: empieza por, no empieza por, termina por, no termina por, contiene, no contiene, es igual, no es igual, coincide, no coincide.

En la siguiente tabla se muestran las estadísticas disponibles que puede utilizar para las estadísticas de distribución de valores y las estadísticas numéricas:

Verificación de la calidad de los datos	Nombre de las estadísticas	Parámetros adicionales	Sintaxis del SDK
Estadísticas de distribución de valores	Mínimo		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	Máximo		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	Median		`"CheckExpression": "AGG(MEDIAN) >= :val", "Substitu`

Verificación de la calidad de los datos	Nombre de las estadísticas	Parámetros adicionales	Sintaxis del SDK
			`tionMap": {":val", "50"}`
	Mean		`"CheckExpression": "AGG(MEAN) <= :val", "SubstitutionMap": {":val", "10"}`
	Mode		`"CheckExpression": "AGG(MODE) > :val", "SubstitutionMap": {":val", "0"}`
	Desviación estándar		`"CheckExpression": "AGG(STANDARD_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`

Verificación de la calidad de los datos	Nombre de las estadísticas	Parámetros adicionales	Sintaxis del SDK
	Entropía		`"CheckExpression": "AGG(ENTROPY) > :val", "SubstitutionMap": {":val", "0"}`
Estadísticas numéricas	Sum		`"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}`
	Curtosis		`"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}`
	Asimetría		`"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}`

Verificación de la calidad de los datos	Nombre de las estadísticas	Parámetros adicionales	Sintaxis del SDK
	Varianza		`"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}`
	Desviación absoluta		`"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Cuantil	Cuantil: uno de «0,25», «0,5», «0,75»	`"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}`

Crear y usar AWS Glue DataBrew proyectos

En AWS Glue DataBrew, un proyecto es la pieza central de sus esfuerzos de análisis y transformación de datos.

Cuando crea un proyecto, reúne dos componentes fundamentales:

- Un conjunto de datos, para proporcionar acceso de solo lectura a los datos de origen. Para obtener más información, consulte Conectarse a los datos con AWS Glue DataBrew.
- Una receta para aplicar transformaciones de DataBrew datos al conjunto de datos. Para obtener más información, consulte Crear y usar AWS Glue DataBrew recipes.

La DataBrew consola presenta su proyecto en una interfaz de usuario intuitiva y altamente interactiva. Le anima a experimentar con cientos de transformaciones de datos para que pueda aprender cómo funcionan y qué efecto tienen en sus datos.

Los datos que ve en la vista del proyecto son una muestra de su conjunto de datos. Como los conjuntos de datos pueden ser muy grandes, con miles o incluso millones de filas, el uso de una muestra ayuda a garantizar que la DataBrew consola mantenga su capacidad de respuesta mientras se transforman los datos de la muestra de diversas formas. De forma predeterminada, la muestra consta de las primeras 500 filas de datos del conjunto de datos. Puede elegir diferentes ajustes para el tamaño de la muestra y las filas que desee.

A medida que transforma los datos de la muestra, le DataBrew ayuda a crear y perfeccionar la receta del proyecto: una serie paso a paso de las transformaciones que ha aplicado hasta ahora. La receta del trabajo en curso se guarda automáticamente, por lo que puede salir de la vista del proyecto en cualquier momento, volver más tarde y continuar donde la dejó.

Cuando la receta esté lista para usarse, podrás publicarla. La publicación de una receta la pone a disposición del subsistema de DataBrew tareas, donde puede aplicarla a todo su conjunto de datos o crear un perfil de datos extenso que le permita comprender la estructura, el contenido y las características estadísticas de los datos.

Temas

- Eliminación de un proyecto

Creación de un proyecto

Utilice el siguiente procedimiento para crear un proyecto.

Para crear un proyecto

1. Inicie sesión en la DataBrew consola Consola de administración de AWS y ábrala.
2. En el panel de navegación, selecciona PROYECTOS. A continuación, elija Crear proyecto.
3. Escriba un nombre para el proyecto. A continuación, elige una receta para adjuntarla a tu proyecto:

   - Elige Crear nueva receta si empiezas desde el principio. Al hacer esto, se crea una receta nueva y vacía y se adjunta al proyecto.
   - Elija Editar una receta existente si tiene una receta publicada anteriormente que desee utilizar para este proyecto. Si la receta está actualmente adjunta a otro proyecto o tiene algún trabajo definido, no podrá utilizarla en el nuevo proyecto. Selecciona Buscar recetas para ver qué recetas están disponibles.
   - Elige Importar pasos de una receta si ya tienes una receta que se ha publicado anteriormente y deseas importar sus pasos, y luego haz lo siguiente:
     1. Selecciona Buscar recetas para ver qué recetas están disponibles.
     2. Elige la versión publicada de la receta que quieres usar. Una receta puede tener varias versiones, según la frecuencia con la que la publiques mientras trabajas en la vista de proyecto.
     3. Seleccione Ver los pasos de la receta para examinar las transformaciones de datos de la receta.
4. Una vez que tenga una receta, elija el conjunto de datos con el que quiere trabajar en el panel Seleccione un conjunto de datos:

   - Mis conjuntos de datos: elija un conjunto de datos que haya creado anteriormente. Para obtener más información, consulte Creación de un proyecto).
   - Archivos de muestra: cree un nuevo conjunto de datos basado en los datos de muestra mantenidos por AWS. Estos datos de muestra son una excelente manera de explorar lo que DataBrew puede hacer sin tener que proporcionar sus propios datos. Asegúrese de introducir un nombre para su conjunto de datos.

- Nuevo conjunto de datos: crea un nuevo conjunto de datos. Para obtener más información, consulte Creación de un proyecto.

5. Para los permisos de acceso, elija un rol AWS Identity and Access Management(IAM) que le permita DataBrew leer desde la ubicación de entrada de Amazon S3. En el caso de una ubicación de S3 propiedad de su AWS cuenta, puede elegir la función gestionada por el `AwsGlueDataBrewDataAccessRole` servicio. De este modo, podrá acceder DataBrew a los recursos de S3 de su propiedad.
6. En el panel de muestreo, puede encontrar opciones para crear una muestra de datos DataBrew a partir de su conjunto de datos.

   En Tipo, elige cómo se DataBrew deben obtener las filas de tu conjunto de datos:

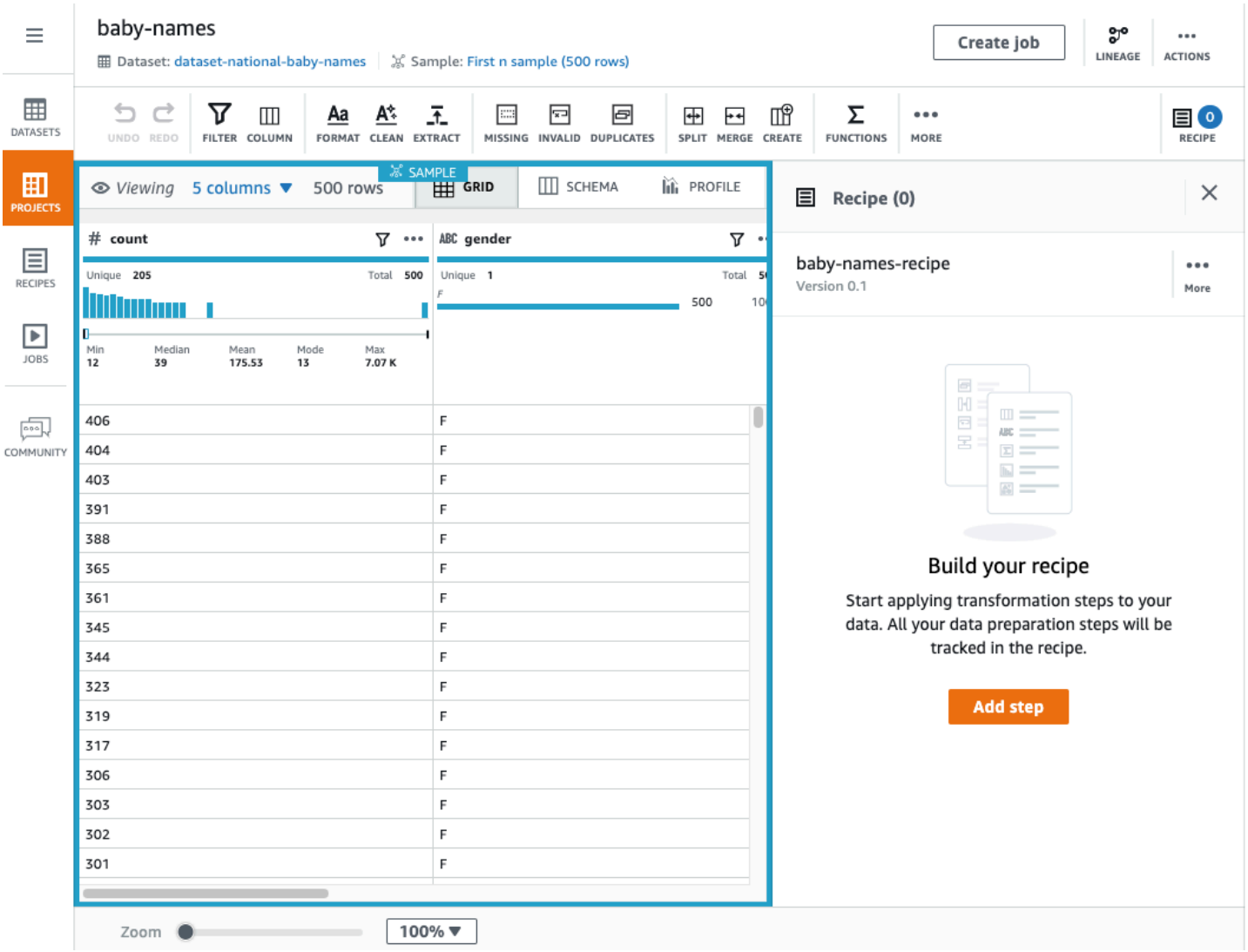
   - Usa First n rows para crear una muestra basada en las primeras filas del conjunto de datos.
   - Use filas aleatorias para crear una muestra basada en una selección aleatoria de filas del conjunto de datos.
   - Elija el número de filas que aparecerán en la muestra: 500, 1000, 2500 o un tamaño de muestra personalizado, hasta un máximo de 5000 filas. Un tamaño de muestra más pequeño permite DataBrew realizar las transformaciones con mayor rapidez, lo que le permite ahorrar tiempo a la hora de elaborar la receta. Un tamaño de muestra más grande refleja con mayor precisión la composición de los datos fuente subyacentes. Sin embargo, la inicialización de la sesión del proyecto y las transformaciones interactivas son más lentas.
7. (Opcional) Elija Etiquetas para adjuntar etiquetas a su conjunto de datos.

   Las etiquetas son etiquetas simples que constan de una clave definida por el usuario y un valor opcional que pueden facilitar la administración, la búsqueda y el filtrado de DataBrew los proyectos por propósito, propietario, entorno u otros criterios.
8. Cuando los ajustes sean los que desea, elija Crear trabajo.

DataBrew crea un nuevo conjunto de datos si es necesario, crea una nueva receta si es necesario, crea la muestra de datos y crea una sesión de proyecto interactiva. Este proceso puede tardar un par de minutos en completarse. Cuando el proyecto esté listo para su uso, puede empezar a trabajar con la muestra de datos.

Descripción general de una sesión de DataBrew proyecto

En una sesión de DataBrew proyecto, trabajas en un espacio de trabajo interactivo.



El panel izquierdo muestra la vista actual de los datos. El panel derecho muestra la receta de transformación del proyecto, que actualmente está vacía.

En la esquina superior derecha de la cuadrícula de datos, hay tres pestañas: `GRID`SCHEMA, y. `PROFILE` Al seleccionar una de estas pestañas, se muestra la vista correspondiente en el espacio de trabajo; estas vistas se describen a continuación.

Vista de cuadrícula

La vista de cuadrícula es la vista por defecto, donde el ejemplo se muestra en formato tabular. Utilice el siguiente procedimiento para obtener un breve recorrido por la vista de cuadrícula.

Para hacer un recorrido por la vista de cuadrícula

1. Comience por ver todo el espacio:

a. Desplázate hacia la izquierda y hacia la derecha para ver todas las columnas.

b. Desplázate hacia arriba y hacia abajo para ver todos los valores de los datos.

c. Utilice el control de zoom en la parte inferior del espacio de trabajo para ajustar el nivel de ampliación de la cuadrícula.

2. En la parte superior derecha, vea cuántas columnas de la muestra se muestran y el número actual de filas de la muestra.

   Para cambiar las columnas que se muestran, elija el enlace N columnas (donde N es el número de columnas que se muestran actualmente). Elija las columnas que desee y elija Mostrar las columnas seleccionadas.

3. Ahora puede empezar a experimentar con DataBrew las transformaciones. Pruebe lo siguiente:

   a. En la barra de herramientas de transformación, elija Elegir formato y cambiar a mayúscula.

   b. En Columna de origen, elija una columna que contenga datos de caracteres.

   c. No cambie los valores predeterminados de los demás ajustes.

   d. Para ver el aspecto que tendrán los datos transformados, selecciona Vista previa de los cambios. A continuación, para añadir esta transformación a la receta, selecciona Aplicar.

   Siempre que aplique una transformación de datos, DataBrew agréguela a la copia de trabajo de la receta. Aparece en el lado derecho de tu espacio de trabajo.

4. Pruebe lo siguiente:

   a. En la barra de herramientas de transformación, selecciona Crear, en función de una función.

   b. En Seleccione una función, elija`SQUARE ROOT`.

   c. En Columna de origen, elija una columna que contenga datos numéricos.

   d. Deje las demás configuraciones en sus valores predeterminados,.

   e. Seleccione Vista previa de los cambios para ver el aspecto de los datos transformados. A continuación, para añadir esta transformación a la receta, selecciona Aplicar.

5. Contraiga el panel de recetas en la esquina superior derecha seleccionando RECETA. Para ampliar el panel de recetas, vuelva a seleccionar RECETA.

Publicar una nueva versión de la receta

A medida que se siguen aplicando las transformaciones, el número de pasos de la receta aumenta. En cualquier momento, puedes publicar una nueva versión de tu receta. La publicación de una receta hace que esté disponible en otros lugares de DataBrew. De este modo, puede ejecutar un trabajo de preparación para transformar todo su conjunto de datos, en lugar de transformar solo la muestra de datos del proyecto.

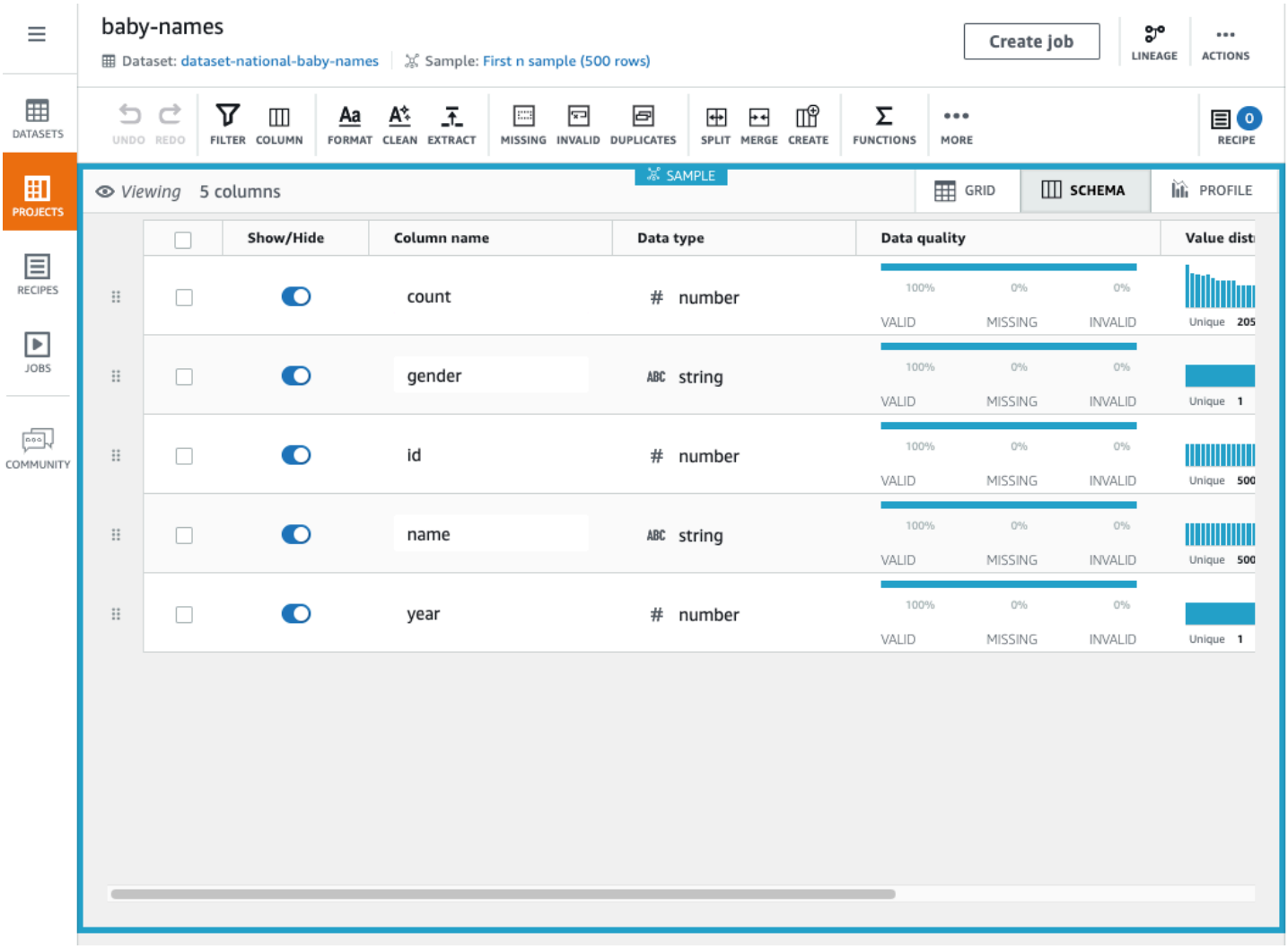
La publicación de recetas también fomenta un enfoque gradual e iterativo para el desarrollo de recetas: puedes publicar nuevas versiones de tu receta a medida que avanzas, de modo que puedes recurrir a la versión «última vez que se sepa que fue buena» si fuera necesario.

Para publicar una nueva versión de una receta

- En el panel de recetas, elija Publicar. Introduzca una descripción para esta versión de la receta y elija Publicar.

Vista de esquema

Si selecciona la pestaña ESQUEMA, la vista cambia, como se muestra en la siguiente captura de pantalla.



En la vista de esquema, puede ver las estadísticas sobre los valores de los datos de cada columna.

En la columna del extremo izquierdo, junto a Show/Hide, elige cualquiera de las columnas de datos. El panel de detalles de la columna aparece a la derecha. Este panel muestra un resumen de las estadísticas de los valores de las columnas.

Puede cambiar el nombre de una columna introduciendo un nombre nuevo para el nombre de la columna.

Puede reorganizar el orden de las columnas arrastrándolas y soltándolas.

Visualización de perfil

Si elige la pestaña PERFIL, puede ver información volumétrica detallada sobre su proyecto. Antes de hacerlo, ejecuta un DataBrew trabajo para crear el perfil.

Para hacer un recorrido por la visualización del perfil

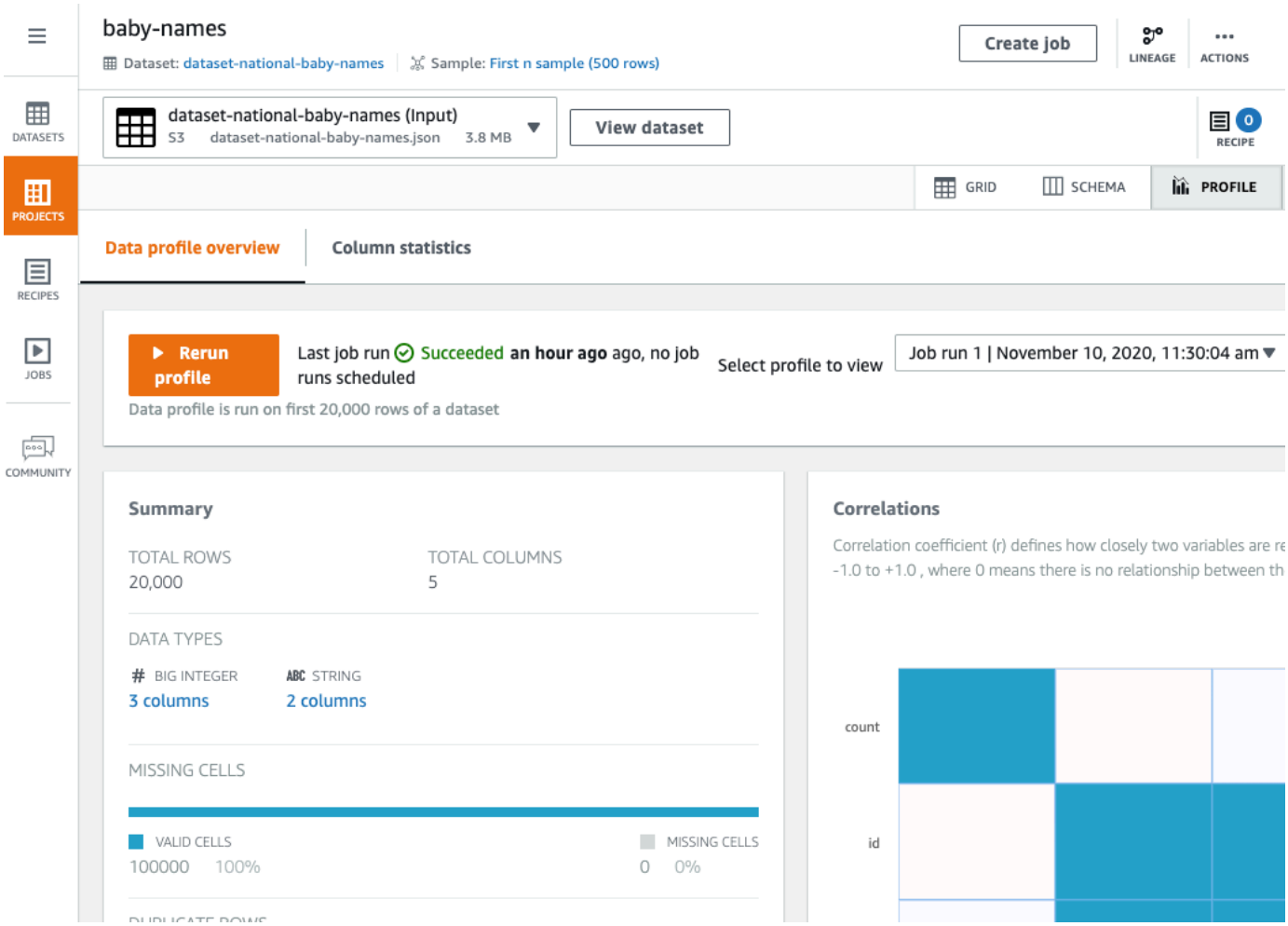
1. Seleccione Crear trabajo e introduzca un nombre para su trabajo.
2. En la salida de Job, elija CSV para el tipo de archivo.
3. Busque o cree un depósito y una carpeta de Amazon S3 en su AWS cuenta en los que desee que se escriba el resultado del DataBrew trabajo:

   - Si ya tiene este bucket y esta carpeta de Amazon S3, seleccione Browse y búsquelos. Asegúrese de tener permisos de escritura para ambos.
   - Si no tienes este bucket y esta carpeta de Amazon S3, créalos:

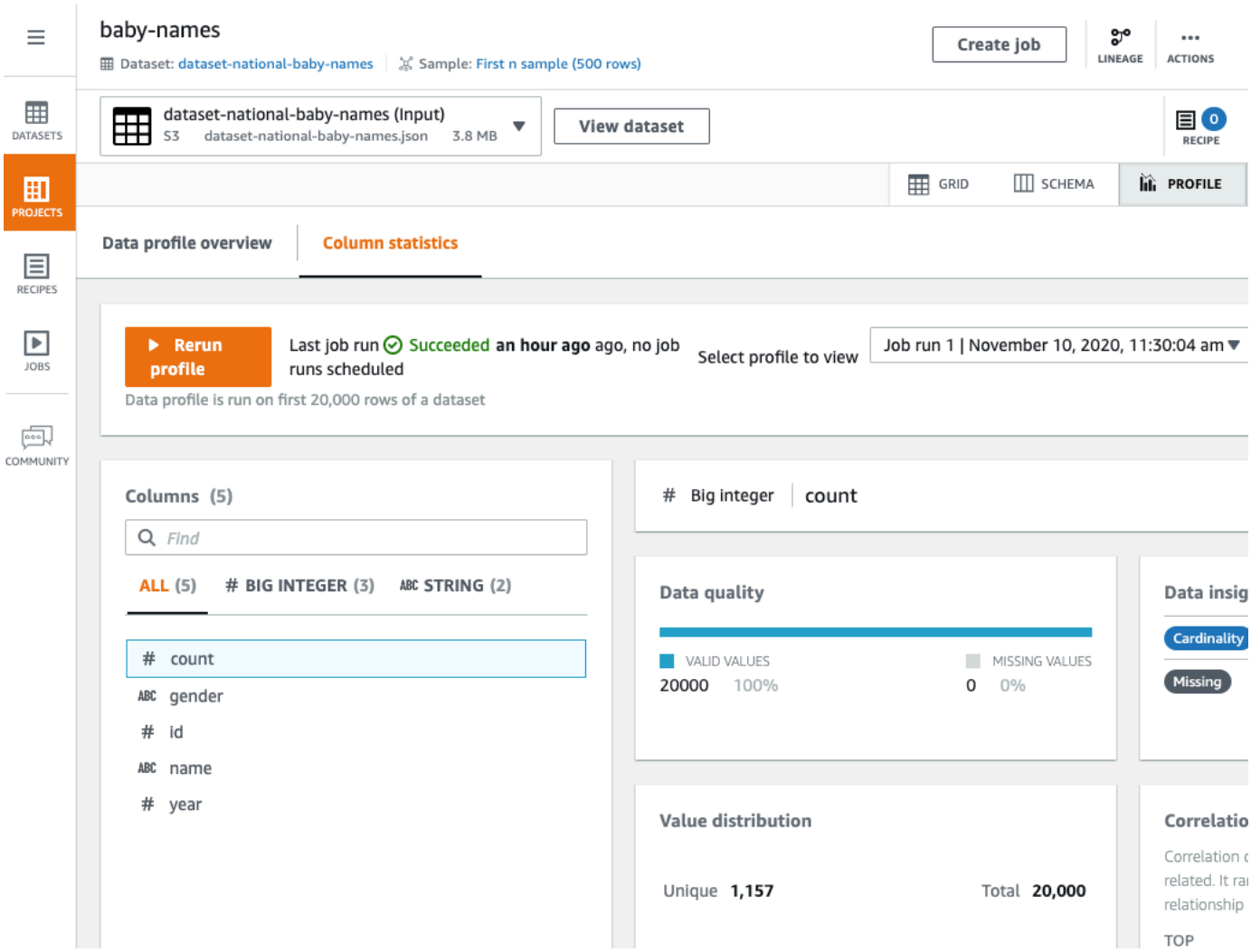
     1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
     2. Si no tiene un bucket de Amazon S3, elija Create bucket. En el campo Nombre del bucket, introduce un nombre exclusivo para el nuevo bucket. Elija Crear bucket.
     3. En la lista de cubos, elige el que quieras usar.
     4. Elija Crear carpeta. En Nombre de carpeta`databrew-output`, introduzca y seleccione Crear carpeta.
4. Para los permisos de acceso, elija un rol de IAM que le permita DataBrew escribir en la ubicación de salida de Amazon S3.

   En el caso de una ubicación de S3 propiedad de su AWS cuenta, puede elegir la función gestionada por el `AwsGlueDataBrewDataAccessRole` servicio. De este modo, podrá acceder DataBrew a los recursos de S3 de su propiedad.
5. Deje las demás configuraciones en sus valores predeterminados y elija Crear y ejecutar trabajo.
6. Una vez finalizada la tarea, el espacio de trabajo muestra un resumen gráfico del perfil de datos.

   La pestaña Resumen del perfil de datos muestra un resumen detallado de las características de los datos, como se muestra en la siguiente captura de pantalla.



La pestaña de estadísticas de columnas muestra un desglose columna por columna de los valores de los datos:



Eliminación de un proyecto

Si ya no necesitas un proyecto, puedes eliminarlo.

Para eliminar un proyecto

1. En el panel de navegación, selecciona PROYECTOS.
2. Elija el proyecto que desee eliminar y, a continuación, en Acciones, elija Eliminar. .

Crear y usar AWS Glue DataBrew recipes

En DataBrew, una receta es un conjunto de pasos de transformación de datos. Puede aplicar estos pasos a una muestra de sus datos o aplicar la misma receta a un conjunto de datos.

La forma más sencilla de desarrollar una receta es crear un DataBrew proyecto, en el que pueda trabajar de forma interactiva con una muestra de sus datos; para obtener más información, consulte. Crear y usar AWS Glue DataBrew proyectos Como parte del flujo de trabajo de creación del proyecto, se crea una receta nueva (vacía) y se adjunta al proyecto. A continuación, puede empezar a crear su receta añadiendo transformaciones de datos.

> **Note**
>
> Puede incluir hasta 100 transformaciones de datos en una sola DataBrew receta.

A medida que vaya desarrollando la receta, puede guardar su trabajo publicándola. DataBrew mantiene una lista de las versiones publicadas de su receta. Puede usar cualquier versión publicada en un trabajo de receta para ejecutar la receta (en un trabajo de receta) y transformar su conjunto de datos. También puedes descargar una copia de los pasos de la receta para reutilizarla en otros proyectos o en otras transformaciones de conjuntos de datos.

También puedes desarrollar DataBrew recetas mediante programación, utilizando AWS Command Line Interface(AWS CLI) o uno de los AWS SDK. En la DataBrew API, las transformaciones se conocen como acciones de receta.

> **Note**
>
> En una sesión de DataBrew proyecto interactiva, cada transformación de datos que se aplique dará como resultado una llamada a la DataBrew API. Estas llamadas a la API se producen automáticamente, sin que tengas que conocer los detalles entre bastidores.

Incluso si no eres programador, es útil entender la estructura de una receta y cómo DataBrew organiza sus acciones.

Temas

- Definir la estructura de una receta

Publicar una nueva versión de la receta

Las nuevas versiones de una receta se publican en una sesión de DataBrew proyecto interactiva.

Para publicar una nueva versión de la receta

1. En el panel de recetas, elija Publicar.
2. Introduzca una descripción para esta versión de la receta y elija Publicar.

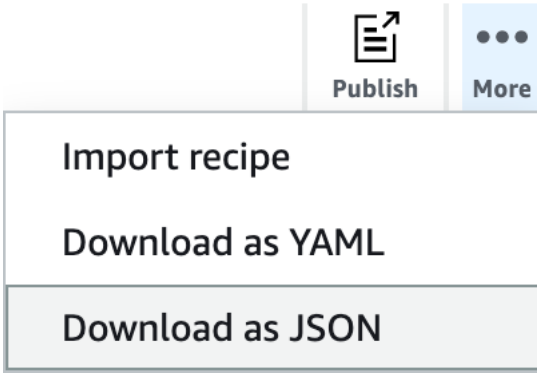
Para ver todas las recetas publicadas y sus versiones, selecciona PROYECTOS en el panel de navegación.

Definir la estructura de una receta

Al crear un proyecto por primera vez con la DataBrew consola, se define una receta para asociarla a ese proyecto. Si no tienes una receta existente, la consola crea una para ti.

Mientras trabaja con el proyecto en la consola, utiliza la barra de herramientas de transformación para aplicar acciones a los datos de muestra del conjunto de datos. La consola muestra los pasos de la receta y su orden a medida que continúa creando la receta. Puede repetir y refinar la receta hasta que esté satisfecho con los pasos.

EnIntroducción al AWS Glue DataBrew, creas una receta para transformar un conjunto de datos de partidas de ajedrez famosas. Para descargar una copia de los pasos de la receta, selecciona Descargar como JSON o Descargar como YAML, como se muestra en la siguiente captura de pantalla.



El archivo JSON descargado contiene las acciones de la receta correspondientes a las transformaciones que has añadido a la receta.

Una receta nueva no tiene ningún paso. Puedes representar una receta nueva como una lista JSON vacía, como se muestra a continuación.

```
[ ]
```

A continuación se muestra un ejemplo de un archivo de este tipo, para `chess-project-recipe`. La lista JSON contiene varios objetos que describen los pasos de la receta. Cada objeto de la lista JSON está entre corchetes ({ }). Las líneas JSON están delimitadas por comas.

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

Es más fácil ver que cada acción es una línea individual si solo añadimos nuevas líneas para las nuevas acciones, como se muestra a continuación.

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

Las acciones se realizan de forma secuencial, en el mismo orden que en el archivo:

- REMOVE_VALUES— Para filtrar todas las partidas en las que la puntuación de un jugador es inferior a 1800, la puntuación mínima requerida para ser un jugador de ajedrez de clase A. Esta acción se puede realizar en dos ocasiones: una para eliminar a los jugadores del lado negro que no sean al menos de clase A, y otra para eliminar a los jugadores del lado blanco que no estén en este nivel.
- GROUP_BY— Para resumir los datos. En este caso, GROUP_BY ordena las filas en grupos en función de los valores de `winner` (`blacky`). `white` A continuación, cada uno de esos grupos se desglosa aún más y las filas se clasifican en subgrupos según los valores de `victory_status` (`mate`, `resignoutoftime`, y). `draw` Por último, se cuenta el número de apariciones de cada subgrupo. A continuación, el resumen resultante reemplaza la muestra de datos original.
- REMOVE_VALUES— Eliminar los resultados de las partidas que terminaron con`draw`.
- REPLACE_TEXT— Para modificar los valores de`victory_status`. Esta acción se repite en tres ocasiones: una para`mate`, `resign` y una para cada una. `outoftime`

En una sesión de DataBrew proyecto interactiva, cada una `RecipeAction` corresponde a una transformación de datos que se aplica a una muestra de datos.

DataBrew proporciona más de 200 acciones de recetas. Para obtener más información, consulte Referencia de pasos y funciones de la receta.

Uso de condiciones

Puedes usar condiciones para limitar el alcance de una acción de receta. Las condiciones se utilizan en las transformaciones que filtran los datos, por ejemplo, al eliminar filas no deseadas en función de un valor de columna concreto.

Echemos un vistazo más de cerca a las acciones de una receta. `chess-project-recipe`

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

```
        "TargetColumn": "black_rating"
      }
    ]
  }
```

Esta transformación lee los valores de la `black_rating` columna. La `ConditionExpressions` lista determina los criterios de filtrado: cualquier fila que tenga un `black_rating` valor inferior a 1800 se elimina del conjunto de datos.

Una transformación de seguimiento en la receta hace lo mismo, para`white_rating`. De esta forma, los datos se limitan a los juegos en los que cada jugador (blanco o negro) tiene una calificación de clase A o superior.

Este es otro ejemplo de una condición, que se aplica a una columna de datos de personajes.

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

Esta transformación lee los valores de la `winner` columna, busca el valor `draw` y elimina esas filas. De esta forma, los datos se limitan únicamente a aquellos juegos en los que hubo un claro ganador.

DataBrew admite las siguientes condiciones:

- `IS`— El valor de la columna es el mismo que el valor que se proporcionó en la condición.
- `IS_NOT`— El valor de la columna no es el mismo que el valor que se proporcionó en la condición.
- `IS_BETWEEN`— El valor de la columna se encuentra entre los `LESS_THAN_EQUAL` parámetros `GREATER_THAN_EQUAL` y.

- `CONTAINS`— El valor de cadena de la columna contiene el valor que se proporcionó en la condición.
- `NOT_CONTAINS`— El valor de la columna no contiene la cadena de caracteres que se proporcionó en la condición.
- `STARTS_WITH`— El valor de la columna comienza con la cadena de caracteres que se proporcionó en la condición.
- `NOT_STARTS_WITH`— El valor de la columna no comienza con la cadena de caracteres que se proporcionó en la condición.
- `ENDS_WITH`— El valor de la columna termina con la cadena de caracteres que se proporcionó en la condición.
- `NOT_ENDS_WITH`— El valor de la columna no termina con la cadena de caracteres que se proporcionó en la condición.
- `LESS_THAN`— El valor de la columna es inferior al valor que se proporcionó en la condición.
- `LESS_THAN_EQUAL`— El valor de la columna es inferior o igual al valor que se proporcionó en la condición.
- `GREATER_THAN`— El valor de la columna es superior al valor que se proporcionó en la condición.
- `GREATER_THAN_EQUAL`— El valor de la columna es mayor o igual al valor que se proporcionó en la condición.
- `IS_INVALID`— El valor de la columna tiene un tipo de datos incorrecto.
- `IS_MISSING`— No hay ningún valor en la columna.

Crear, ejecutar y programar AWS Glue DataBrew jobs

AWS Glue DataBrew tiene un subsistema de trabajos que cumple dos propósitos:

1. Aplicar una receta de transformación de datos a un DataBrew conjunto de datos. Esto se hace con una DataBrew receta.
2. Analizar un conjunto de datos para crear un perfil completo de los datos. Esto se hace con un trabajo DataBrew de perfil.

Temas

Crear y trabajar con AWS Glue DataBrew trabajos de recetas

Utilice una DataBrew receta para limpiar y normalizar los datos de un DataBrew conjunto de datos y escriba el resultado en la ubicación de salida que elija. La ejecución de un trabajo de preparación no afecta al conjunto de datos ni a los datos de origen subyacentes. Cuando se ejecuta un trabajo, se conecta a los datos de origen en modo de solo lectura. El resultado del trabajo se escribe en una ubicación de salida que usted defina en Amazon S3, en la AWS Glue Data Catalog base de datos JDBC compatible o en una base de datos JDBC compatible.

Utilice el siguiente procedimiento para crear un trabajo de DataBrew receta.

Para crear un trabajo de recetas

1. Inicie sesión en Consola de administración de AWS y abra la DataBrew consola en https://console.aws.amazon.com/databrew/.
2. Elija TRABAJOS en el panel de navegación, elija la pestaña Recipe trabajos y, a continuación, elija Crear trabajo.
3. Introduce un nombre para tu trabajo y, a continuación, selecciona Crear un trabajo de recetas.
4. En la entrada de trabajo, introduzca los detalles del trabajo que desea crear: el nombre del conjunto de datos que se va a procesar y la receta que se va a utilizar.

   Un trabajo de receta usa una DataBrew receta para transformar un conjunto de datos. Para usar una receta, asegúrate de publicarla primero.

5. Configure los ajustes de salida de sus trabajos.

   Proporcione un destino para el resultado de su trabajo. Si no tiene una DataBrew conexión configurada para su destino de salida, configúrela primero en la pestaña CONJUNTOS DE DATOS, tal y como se describe enConexiones compatibles para fuentes y salidas de datos. Elija uno de los siguientes destinos de salida:

   - Amazon S3, con o sin AWS Glue Data Catalog soporte
   - Amazon Redshift, con o sin soporte AWS Glue Data Catalog
   - JDBC
   - Mesas Snowflake
   - Tablas de bases de datos de Amazon RDS con AWS Glue Data Catalog soporte. Las tablas de bases de datos de Amazon RDS admiten los siguientes motores de bases de datos:
     - Amazon Aurora
     - MySQL
     - Oracle
     - PostgreSQL
     - Microsoft SQL Server
   - Amazon S3 con AWS Glue Data Catalog soporte.

   Para la AWS Glue Data Catalog salida basada en AWS Lake Formation, solo DataBrew admite la sustitución de los archivos existentes. En este enfoque, los archivos se sustituyen para mantener intactos sus permisos actuales de Lake Formation para su función de acceso a los datos. Además, DataBrew da prioridad a la ubicación de Amazon S3 de la AWS Glue Data Catalog tabla. Por lo tanto, no puede anular la ubicación de Amazon S3 al crear un trabajo de receta.

   En algunos casos, la ubicación de Amazon S3 en el resultado del trabajo difiere de la ubicación de Amazon S3 en la tabla del catálogo de datos. En estos casos, DataBrew actualiza la definición del trabajo automáticamente con la ubicación de Amazon S3 de la tabla del catálogo. Lo hace cuando actualiza o inicia sus trabajos existentes.

6. Solo para los destinos de salida de Amazon S3, tiene más opciones:

   a. Elija uno de los formatos de salida de datos disponibles para Amazon S3, la compresión opcional y un delimitador personalizado opcional. Los delimitadores admitidos para los

archivos de salida son los mismos que para los de entrada: coma, dos puntos, punto y coma, barra vertical, tabulación, intercalación, barra invertida y espacio. Para obtener información sobre el formato, consulte la siguiente tabla.

Formato	Extensión de archivo (sin comprimir)	Extensiones de archivo (comprimidas)
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	No compatible	`.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`

Formato	Extensión de archivo (sin comprimir)	Extensiones de archivo (comprimidas)
JSON (solo en formato de líneas JSON)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	No compatible	No aplicable

b. Elija si desea generar un solo archivo o varios archivos. Existen tres opciones para la salida de archivos con Amazon S3:

- Generar archivos automáticamente (recomendado): ha DataBrew determinado el número óptimo de archivos de salida.
- Salida de un solo archivo: hace que se genere un solo archivo de salida. Esta opción puede provocar un tiempo adicional de ejecución del trabajo, ya que es necesario realizar un posprocesamiento.
- Salida de varios archivos: permite especificar el número de archivos para la salida del trabajo. Los valores válidos son de 2 a 999. Es posible que se generen menos archivos de los que especifique si se utiliza la partición de columnas o si el número de filas de la salida es inferior al número de archivos que especifique.

c. (Opcional) Elija la partición de columnas para la salida del trabajo de receta.

La partición de columnas proporciona otra forma de dividir el resultado del trabajo de receta en varios archivos. La partición de columnas se puede utilizar con la salida de Amazon S3 nueva o existente o con la salida del nuevo catálogo de datos de Amazon S3. No se puede usar con las tablas del catálogo de datos Amazon S3 existentes. Los archivos de salida se basan en los valores de los nombres de columna que especifique. Si los nombres de columna que especifique son únicos, las rutas de las carpetas de Amazon S3 resultantes se basan en el orden de los nombres de las columnas.

Para ver un ejemplo de partición de columnasEjemplo de partición de columnas, consulte lo siguiente.

7. (Opcional) Seleccione Activar el cifrado de la salida del trabajo para cifrar la salida del trabajo que se DataBrew escribe en la ubicación de salida y, a continuación, elija el método de cifrado:

- Utilice SSE-S3 cifrado: la salida se cifra mediante cifrado del lado del servidor con claves de cifrado administradas por Amazon S3.
- Use AWS Key Management Service(AWS KMS): la salida se cifra mediante.AWS KMS Para usar esta opción, elija el nombre de recurso de Amazon (ARN) de la AWS KMS clave que desee usar. Si no tiene una AWS KMS clave, puede crear una seleccionando Crear una AWS KMS clave.

8. Para los permisos de acceso, elige un rol AWS Identity and Access Management(IAM) que te permita DataBrew escribir en tu ubicación de salida. En el caso de una ubicación que sea propiedad de tu AWS cuenta, puedes elegir la función de administración del `AwsGlueDataBrewDataAccessRole` servicio. De este modo, podrá acceder DataBrew a AWS los recursos de su propiedad.
9. En el panel de configuración avanzada del trabajo, puede elegir más opciones para ejecutar el trabajo:

   - Número máximo de unidades: DataBrew procesa los trabajos con varios nodos de cómputo y se ejecutan en paralelo. El número predeterminado de nodos es 5. El número máximo de nodos es 149.
   - Tiempo de espera del trabajo: si un trabajo tarda más de los minutos que ha establecido aquí en ejecutarse, se produce un error de tiempo de espera. El valor predeterminado es de 2880 minutos o 48 horas.
   - Número de reintentos: si un trabajo falla mientras se está ejecutando, DataBrew puede intentar volver a ejecutarlo. De forma predeterminada, el trabajo no se vuelve a intentar.
   - Habilitar Amazon CloudWatch Logs para el trabajo: DataBrew permite publicar información de diagnóstico en CloudWatch Logs. Estos registros pueden ser útiles para solucionar problemas o para obtener más detalles sobre cómo se procesa el trabajo.

10. En el caso de los trabajos programados, puede aplicar un DataBrew cronograma de trabajo para que el trabajo se ejecute en un momento determinado o de forma periódica. Para obtener más información, consulte Automatizar la ejecución de los trabajos con un cronograma.
11. Cuando los ajustes sean los que desea, elija Crear trabajo. O bien, si desea ejecutar el trabajo inmediatamente, elija Crear y ejecutar trabajo.

Puede supervisar el progreso del trabajo comprobando su estado mientras el trabajo está en ejecución. Cuando se completa la ejecución del trabajo, el estado cambia a Se ha realizado correctamente. La salida del trabajo ahora está disponible en la ubicación de salida elegida.

DataBrew guarda la definición del trabajo para que pueda ejecutar el mismo trabajo más adelante. Para volver a ejecutar un trabajo, seleccione Trabajos en el panel de navegación. Elija el trabajo con el que quiere trabajar y, a continuación, elija Ejecutar trabajo.

Ejemplo de partición de columnas

Como ejemplo de partición de columnas, supongamos que se especifican tres columnas, cada una de las cuales contiene uno de los dos valores posibles. La `Dept` columna puede tener el valor `Admin` o`Eng`. La `Staff-type` columna puede tener el valor `Part-time` o`Full-time`. La `Location` columna puede tener el valor `Office1` o`Office2`. Los buckets de Amazon S3 para la producción de su trabajo tienen un aspecto similar al siguiente.

```
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
```

Automatizar la ejecución de los trabajos con un cronograma

Puede volver a ejecutar los DataBrew trabajos en cualquier momento y también automatizar las ejecuciones de los DataBrew trabajos con un cronograma.

Para volver a ejecutar un trabajo DataBrew

1. Inicie sesión en Consola de administración de AWS y abra la DataBrew consola en https://console.aws.amazon.com/databrew/.
2. En el panel de navegación, selecciona Trabajos. Elija el trabajo que desee ejecutar y, a continuación, elija Ejecutar trabajo.

Para ejecutar un DataBrew trabajo en un momento determinado o de forma periódica, cree un cronograma de DataBrew trabajos. A continuación, puede configurar su trabajo para que se ejecute de acuerdo con el cronograma.

Para crear un cronograma de DataBrew tareas

1. En el panel de navegación de la DataBrew consola, elija Trabajos. Seleccione la pestaña Programaciones y, a continuación, seleccione Añadir programación.
2. Introduce un nombre para tu programación y, a continuación, elige un valor para la frecuencia de ejecución:

   - Recurrente: elija la frecuencia con la que desea que se ejecute el trabajo (por ejemplo, cada 12 horas). A continuación, elija el día o los días en los que desea ejecutar el trabajo. Si lo desea, puede introducir la hora del día en que se ejecuta el trabajo.
   - A una hora determinada: introduzca la hora del día en la que desea que se ejecute el trabajo. A continuación, elija el día o los días en los que desea ejecutar el trabajo.
   - Introduzca CRON: defina el horario de trabajo introduciendo una expresión cron válida. Para obtener más información, consulte Trabaje con expresiones cron para trabajos de elaboración de recetas.
3. Cuando la configuración sea la que desea, elija Save (Guardar).

Para asociar un trabajo a un cronograma

1. En el panel de navegación, elija Trabajos.
2. Elija el trabajo con el que quiere trabajar y, a continuación, en Acciones, elija Editar. .
3. En el panel Programar trabajos, seleccione Asociar programación. Elija el nombre de la programación que desee usar.
4. Cuando la configuración sea la que desea, elija Save (Guardar).

Trabaje con expresiones cron para trabajos de elaboración de recetas

Las expresiones Cron tienen seis campos obligatorios, que están separados por un espacio en blanco. La sintaxis es la siguiente.

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

En la sintaxis anterior, se utilizan los siguientes valores y caracteres comodín para los campos indicados.

Campos	Valores	Caracteres comodín
Minutos	0–59	, - * /
Horas	0–23	, - * /
Day-of-month	1–31	, - * ? / L W
Mes	1—12 o JAN-DEC	, - * /
Day-of-week	1—7 o SUN-SAT	, - * ? / L
Año	1970-2199	, - * /

Utilice estos caracteres comodín de la siguiente manera:

- El carácter comodín , (coma) incluye valores adicionales. En el `Month` campo, `JAN,FEB,MAR` incluye enero, febrero y marzo.
- El comodín - (al trazo) especifica los rangos. En el `Day` campo, del 1 al 15 se incluyen los días 1 a 15 del mes especificado.
- El * (asterisco) incluye todos los valores del campo. En el `Hours` campo, * incluye todas las horas.
- El comodín / (barra inclinada) especifica incrementos. En el `Minutes` campo, puede **`1/10`** especificar cada 10 minutos, empezando por el primer minuto de la hora (por ejemplo, los minutos 11, 21 y 31).
- El comodín ? (signo de interrogación) especifica uno u otro. Por ejemplo, supongamos que en el `Day-of-month` campo introduce 7. Si no le importa qué día de la semana es el séptimo, ¿puede escribir? en el `Day-of-week` campo.
- El comodín L del `Day-of-week` campo `Day-of-month` o especifica el último día del mes o de la semana.
- El comodín W en el campo `Day-of-month` especifica un día de la semana. En el campo `Day-of-month`, `3W` especifica el día más cercano al tercer día de semana del mes.

Estos campos y valores tienen las siguientes limitaciones:

- No se pueden especificar los campos `Day-of-month` y `Day-of-week` en la misma expresión Cron. Si especifica un valor en uno de los campos, debe utilizar un ? (signo de interrogación) en el otro.
- No se admiten las expresiones cron que generan velocidades superiores a 5 minutos.

Cuando cree una programación, puede utilizar las siguientes cadenas Cron de ejemplo.

Minutos	Horas	Día del mes	Mes	Día de la semana	Año	Significado
0	10	*	*	?	*	Corre a las 10:00 a. m. (UTC) todos los días
15	12	*	*	?	*	Ejecutar a las 12:15 (UTC) todos los días
0	18	?	*	MON-FRI	*	Ejecutar a las 18:00 (UTC) de lunes a viernes
0	8	1	*	?	*	Corre a las 8:00 a. m. (UTC) el primer día del mes
0/15	*	*	*	?	*	Ejecutar cada 15 minutos

Minutos	Horas	Día del mes	Mes	Día de la semana	Año	Significado
0/10	*	?	*	MON-FRI	*	Ejecutar cada 10 minutos de lunes a viernes
0/5	8-17	?	*	MON-FRI	*	Ejecutar cada 5 minutos de lunes a viernes entre las 8:00 y las 17:55 (UTC)

Por ejemplo, puedes usar la siguiente expresión cron para ejecutar un trabajo todos los días a las 12:15 UTC.

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

Eliminar trabajos y cronogramas de trabajos

Si ya no necesita un trabajo o un cronograma de trabajo, puede eliminarlos.

Eliminación de un trabajo

1. En el panel de navegación, elija Trabajos.
2. Elija el trabajo que desee eliminar y, a continuación, en Acciones, elija Eliminar. .

Para eliminar un cronograma de trabajo

1. En el panel de navegación, elija Trabajos y, a continuación, elija la pestaña Programaciones.
2. Elija la programación que desee eliminar y, a continuación, en Acciones, elija Eliminar. .

Crear y trabajar con AWS Glue DataBrew trabajos de perfil

Los trabajos de perfil ejecutan una serie de evaluaciones en un conjunto de datos y envían los resultados a Amazon S3. La información que recopila la creación de perfiles de datos le ayuda a comprender su conjunto de datos y a decidir qué tipo de pasos de preparación de datos desea seguir en sus trabajos de preparación de datos.

La forma más sencilla de ejecutar un trabajo de perfil es utilizar la configuración predeterminada DataBrew . Puede configurar el trabajo de perfil antes de ejecutarlo para que devuelva solo la información que desee.

Utilice el siguiente procedimiento para crear un trabajo DataBrew de perfil.

Para crear un trabajo de perfil

1. Inicie sesión en Consola de administración de AWS y abra la DataBrew consola en https://console.aws.amazon.com/databrew/.
2. Elija TRABAJOS en el panel de navegación, elija la pestaña Perfil de trabajos y, a continuación, elija Crear trabajo.
3. Introduzca un nombre para su trabajo y, a continuación, seleccione Crear un trabajo de perfil.
4. Para la entrada de Job, proporcione el nombre del conjunto de datos que se va a perfilar.
5. (Opcional) Configure lo siguiente en el panel de configuraciones del perfil de datos:

   - Configuraciones a nivel de conjunto de datos: configure los detalles del trabajo de su perfil para todas las columnas de su conjunto de datos.

     Si lo desea, puede activar la capacidad de detectar y contar filas duplicadas en el conjunto de datos. También puedes elegir Activar la matriz de correlaciones y seleccionar columnas para ver qué tan estrechamente están relacionados los valores de varias columnas. Para obtener más información sobre las estadísticas que puede configurar a nivel de conjunto de datos, consulteEstadísticas configurables a nivel de conjunto de datos. Puede configurar las estadísticas en la DataBrew consola o mediante la DataBrew API o AWS los SDK.
   - Configuraciones a nivel de columna: con los ajustes de configuración de perfil predeterminados, puede seleccionar las columnas que desee incluir en el trabajo de su perfil. Utilice Añadir anulación de configuración para seleccionar las columnas cuyas columnas desea limitar el número de estadísticas recopiladas o anular la configuración predeterminada de determinadas estadísticas. Para obtener información detallada sobre las estadísticas que

puede configurar a nivel de columna, consulte. Estadísticas configurables a nivel de columna Puede configurar las estadísticas en la DataBrew consola o mediante la DataBrew API o AWS los SDK.

Asegúrese de que cualquier modificación de configuración que especifique se aplique a las columnas que haya incluido en el trabajo de su perfil. Si hay conflictos entre las distintas anulaciones que haya configurado para una columna, tendrá prioridad la última modificación conflictiva.

6. (Opcional) Puede crear reglas de calidad de los datos y aplicar conjuntos de reglas adicionales asociados a este conjunto de datos o eliminar los que ya se hayan aplicado. Para obtener más información sobre la validación de la calidad de los datos, consulte. Validación de la calidad de los datos en AWS Glue DataBrew
7. En el panel de configuración avanzada del trabajo, puede elegir más opciones para ejecutar el trabajo:

   - Número máximo de unidades: DataBrew procesa los trabajos con varios nodos de cómputo y se ejecutan en paralelo. El número predeterminado de nodos es 5. El número máximo de nodos es 149.
   - Tiempo de espera del trabajo: si un trabajo tarda más de los minutos que ha establecido aquí en ejecutarse, se produce un error de tiempo de espera. El valor predeterminado es de 2880 minutos o 48 horas.
   - Número de reintentos: si un trabajo falla mientras se está ejecutando, DataBrew puede intentar volver a ejecutarlo. De forma predeterminada, el trabajo no se vuelve a intentar.
   - Habilitar Amazon CloudWatch Logs para el trabajo: DataBrew permite publicar información de diagnóstico en CloudWatch Logs. Estos registros pueden ser útiles para solucionar problemas o para obtener más detalles sobre cómo se procesa el trabajo.

8. En el caso de Associated Schedule, puede aplicar un DataBrew cronograma de trabajo para que el trabajo se ejecute en un momento determinado o de forma periódica. Para obtener más información, consulte Automatizar la ejecución de los trabajos con un cronograma.
9. Cuando los ajustes sean los que desea, elija Crear trabajo. O bien, si desea ejecutar el trabajo inmediatamente, elija Crear y ejecutar trabajo.

Crear una configuración de trabajo de perfil mediante programación en AWS Glue DataBrew

En esta sección, encontrará descripciones de los pasos y funciones de los trabajos de perfil que puede utilizar mediante programación. Puede utilizarlos desde AWS Command Line Interface(AWS CLI) o mediante uno de los AWS SDK.

En un trabajo de perfil, puedes personalizar una configuración para controlar cómo se DataBrew evalúa tu conjunto de datos. Puede aplicar la configuración a un conjunto de datos o aplicarla a columnas concretas. Puede crear la configuración al crear un trabajo de perfil y, a continuación, actualizarla en cualquier momento.

La estructura de configuración de un perfil incluye cuatro partes:

- ProfileColumns sección
- DatasetStatisticsConfiguration sección
- ColumnStatisticsConfigurations sección
- EntityDetectorConfiguration sección para configurar la PII

A continuación se muestra un ejemplo.

```
{
    "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 sección

En la `ProfileColumns` sección de tu estructura, establece las columnas del conjunto de datos que deseas evaluar en tu puesto de perfil. `ProfileColumns`ses una lista de selectores de columnas (`Selectors`). Puede especificar un nombre de columna o una expresión regular en un selector de columnas. Ejemplo:

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

Si `ProfileColumns` se especifica, solo se incluyen en el trabajo de perfil las columnas cuyos nombres `ProfileColumns` coincidan con un nombre o una expresión regular. Si el trabajo del perfil

no admite el tipo de datos de una columna seleccionada, DataBrew omite la columna seleccionada durante la ejecución del trabajo.

Si no ProfileColumns está definido, el trabajo del perfil evalúa todas las columnas compatibles. Las columnas compatibles son columnas que contienen datos de un tipo de datos compatible: `ByteTypeShortType`,`IntegerType`,`LongType`,, `FloatType` `DoubleTypeString`, o. `Boolean`

DatasetStatisticsConfiguration sección

En la `DatasetStatisticsConfiguration` sección de su estructura, puede crear una configuración para las evaluaciones entre columnas. La configuración incluye `IncludedStatistics` y. `Overrides` Ejemplo:

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

Puede seleccionar las evaluaciones que desee tener añadiendo nombres a las evaluaciones`IncludedStatistics`. Ejemplo:

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

Si lo especifica`IncludedStatistics`, solo las evaluaciones de la lista se incluyen en el trabajo del perfil. Si no `IncludedStatistics` está definido, el trabajo de perfil ejecuta todas las evaluaciones compatibles con la configuración predeterminada. Puede excluir todas las evaluaciones añadiendo NINGUNA a`IncludedStatistics`. Ejemplo:

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

Estadísticas configurables a nivel de conjunto de datos

En la `DatasetStatisticsConfiguration` sección de su estructura, un puesto de perfil respalda las evaluaciones que se muestran en la siguiente tabla.

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
DUPLICATE_ROWS_COUNT	Recuento de filas duplicadas en el conjunto de datos	all	Habilitado	duplicadoRowsCoun	Int
CORRELACIÓN	Coeficiente de correlación de Pearson entre dos columnas	número	Habilitado	correlaciones (en cada columna seleccionada)	Objeto

En`IncludedStatistics`, puede anular la configuración predeterminada de cada evaluación añadiendo una anulación. Cada anulación incluye el nombre de una evaluación concreta y un mapa de parámetros.

En`DatasetStatisticsConfiguration`, un trabajo de perfil admite la `CORRELATION` anulación. Esta anulación calcula el coeficiente de correlación de Pearson entre dos columnas de una lista de columnas seleccionadas. La configuración predeterminada es seleccionar las 10 primeras columnas numéricas. Puede especificar un número de columnas o una lista de selectores de columnas para anular la configuración predeterminada.

`CORRELATION`toma estos parámetros:

- `columnNumber`— El número de columnas numéricas. El trabajo de perfil selecciona las n primeras columnas del conjunto de datos. Este valor debe ser superior a 1. Se utiliza `"ALL"` para seleccionar todas las columnas numéricas.
- `columnSelectors:`— Lista de selectores de columnas. Cada selector puede tener un nombre de columna o una expresión regular.

Ejemplo:

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

ColumnStatisticsConfigurations sección

En la `ColumnStatisticsConfigurations` sección de su estructura, puede crear configuraciones para columnas específicas. `ColumnStatisticsConfigurations`es una lista de `ColumnStatisticsConfiguration` ajustes. En `ColumnStatisticsConfiguration` ella hay `Selectors` una lista de selectores de columnas y `Statistics` para la configuración de las estadísticas. Ejemplo:

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

`Selectors`es una lista de selectores de columnas. Al igual que en este caso`ProfileColumns`, puede especificar un nombre de columna o una expresión regular en cada selector de columnas. Si lo especifica`Selectors`, la configuración de columnas se aplica a las columnas que coincidan con cualquier selector de columnas`Selectors`. De lo contrario, la configuración se aplica a todas las columnas compatibles.

En`Statistics`, puede anular la configuración de las columnas seleccionadas. Al igual que con`DatasetStatisticsConfiguration`, `Statistics` tiene `IncludedStatistics` y. `Overrides`

Para seleccionar las evaluaciones que desee, añada los nombres de las evaluaciones a`IncludedStatistics`.

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

Si lo especifica`IncludedStatistics`, solo las evaluaciones de la lista se incluyen en el trabajo del perfil. De lo contrario, el trabajo de perfil ejecuta todas las evaluaciones compatibles con la configuración predeterminada.

Puede excluir todas las evaluaciones añadiendo `NONE` a`IncludedStatistics`.

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

En algunos casos, es posible `ColumnStatisticsConfigurations` que haya varias configuraciones diferentes `IncludedStatistics` que se puedan aplicar a la misma columna. En estos casos, el trabajo de perfil selecciona la última configuración `ColumnStatisticsConfigurations` y la aplica `IncludedStatistics` a la columna seleccionada. Una nueva configuración anula las configuraciones anteriores.

Estadísticas configurables a nivel de columna

En`ColumnStatisticsConfigurations`, un trabajo de perfil respalda las evaluaciones que se muestran en la siguiente tabla.

Un tipo de datos admitido `number` en esta tabla significa que el tipo de datos del atributo es uno de los siguientes: `ByteType`,`ShortType`,`IntegerType`,`LongType`,`FloatType`, o`DoubleType`.

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
–	El nombre de la columna.	all	–	name	cadena
–	Tipo de datos de la columna.	all	–	type	cadena
CONTEO DE VALORES DISTINTIVOS	Número de valores distintos. Un valor distinto es un valor que aparece al menos una vez.	number/boolean/cadena	Habilitado	distintoValuesCount	Int
ENTROPÍA	Entropía (teoría de la información).	number/boolean/cadena	Habilitado	entropía	Double
RANGO INTER_CUARTIL	Oscila entre el 25 y el 75 por ciento de los números.	número	Habilitado	Rango intercuartil	Double
CURTOSIS	Curtosis de la columna.	número	Habilitado	curtosis	Double
MAX	Valor máximo de la columna.	number/string longitud	Habilitado	max	Int/Double
VALORES_MÁXIMOS	Lista de los valores máximos de la columna y sus recuentos.	número	Habilitado	Valores máximos	Enumeración
MEAN	Valor medio de los valores de la columna.	number/string longitud	Habilitado	mean	Double

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
MEDIAN	Mediana de los valores de la columna.	number/string longitud	Habilitado	median	Double
DESVIACIÓN ABSOLUTA MEDIA	La mediana de las diferencias absolutas entre cada punto de datos y la mediana de una columna numérica.	número	Habilitado	medianaAbsoluteDeviation	Double
MIN	Valor mínimo de la columna.	number/string longitud	Habilitado	min	Int/Double
VALORES_MÍNIMOS	Lista de los valores mínimos de la columna y sus recuentos.	número	Habilitado	Valores mínimos	Enumeración
CONTEO DE VALORES FALTANTES	Número de valores faltantes en la columna. Las cadenas nulas y vacías se consideran ausentes.	all	Habilitado	faltaValuesCount	Int

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
MODE	El valor que aparece con más frecuencia en la columna. Si varios valores aparecen con esa frecuencia, la moda es uno de esos valores.	number/string longitud	Habilitado	mode	Int/Double
VALORES MÁS COMUNES	Lista de los valores más comunes de la columna.	number/boolean/cadena	Habilitado	la mayoría CommonVa ues	Enumeración
DETECCIÓN DE VALORES ATÍPICOS	Detecta valores atípicos en la columna mediante el algoritmo Z_score. Cuente el número de valores atípicos y extraiga una lista de muestras de los valores atípicos detectados.	number/string longitud	Habilitado	zScoreOutliersCount, zScoreOutliersSample	Int/List

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
PERCENTILES	Valores percentiles de la columna numérica (5%, 25%, 75%, 95%).	número	Habilitado	percentil 5, percentil 25, percentil 75, percentil 95	Double
RANGE	Rango de valores de la columna.	número	Habilitado	rango	Int/Double
ASIMETRÍA	Asimetría de los valores de la columna.	número	Habilitado	asimetría	Double
DESVIACIÓN_ESTÁNDAR	Desviación estándar muestral imparcial de los valores de la columna.	number/string longitud	Habilitado	Desviación estándar	Double
SUM	Suma de los valores de la columna.	número	Habilitado	sum	Int/Double
CONTEO DE VALORES ÚNICOS	Número de valores únicos. Un valor único significa que el valor aparece solo una vez.	number/boolean/cadena	Habilitado	único ValuesCount	Int

Nombre de la estadística	Descripción	Tipos de datos compatibles	Estado predeterminado	Atributos del resultado del perfil	Tipo de resultado del perfil
DISTRIBUCIÓN DE VALORES	Medida de la distribución de los valores de la columna por rango.	number/string longitud	Habilitado	Distribución de valores	Enumeración
VARIANCE	Varianza de los valores de la columna.	número	Habilitado	variance	Double
Z_SCORE_DISTRIBUTION	Medida de la distribución de los valores de puntuación z de los puntos de datos por rango.	número	Habilitado	z ScoreDistribution	Enumeración
ZEROS_COUNT	Número de ceros (0s) en la columna.	número	Habilitado	Zero/ Count	Int

En`IncludedStatistics`, puede anular los parámetros predeterminados de cada evaluación añadiendo una anulación. Cada anulación incluye el nombre de una evaluación concreta y un mapa de parámetros.

Parámetros de las columnas ColumnStatisticsConfigurations

En`ColumnStatisticsConfigurations`, un trabajo de perfil admite los siguientes parámetros.

En algunos casos, es posible `ColumnStatisticsConfigurations` que haya varias configuraciones diferentes `IncludedStatistics` que se puedan aplicar a la misma columna. En estos casos, el trabajo de perfil selecciona la última configuración `ColumnStatisticsConfigurations` y la aplica `IncludedStatistics` a la columna seleccionada. Una nueva configuración anula las configuraciones anteriores.

VALORES_MÁXIMOS

Muestra los valores máximos de la columna numérica y sus recuentos. El tamaño predeterminado de la lista es 5. Puede anular el tamaño de la lista especificando un valor para`sampleSize`.

Configuración

`sampleSize`— El tamaño de la lista que incluye el número máximo y el recuento de valores de la columna numérica. Este valor debe ser mayor que 0. Se utiliza `"ALL"` para enumerar todos los valores.

Ejemplo

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

VALORES_MÍNIMOS

Muestra los valores mínimos de la columna numérica y sus recuentos. El tamaño predeterminado de la lista es 5. Puede anular el tamaño de la lista especificando un valor para`sampleSize`.

Configuración

`sampleSize`— El tamaño de la lista que incluye el número máximo y el recuento de valores de la columna numérica. Este valor debe ser mayor que 0. Se utiliza `"ALL"` para enumerar todos los valores.

Ejemplo

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

LOS VALORES MÁS COMUNES

Muestra los valores más comunes de la columna y sus recuentos. El tamaño predeterminado de la lista es 50. Puede anular el tamaño de la lista especificando un valor para`sampleSize`.

Configuración

`sampleSize`— El tamaño de la lista que incluye el número máximo y el recuento de valores de la columna numérica. Este valor debe ser mayor que 0. Se utiliza `"ALL"` para enumerar todos los valores.

Ejemplo

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

DETECCIÓN DE VALORES ATÍPICOS

Detecta los valores atípicos en la columna numérica o en la columna de cadenas (en función de la longitud de la cadena) mediante el algoritmo z_Score.

Su trabajo de perfil cuenta el número de valores atípicos y genera una lista de muestra de valores atípicos y sus puntuaciones z. La lista de muestras está ordenada por el valor absoluto de la puntuación z. El tamaño predeterminado de la lista es 50.

El algoritmo Z_Score identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar. El umbral de valores atípicos predeterminado es 3.

Puede proporcionar un umbral más, un umbral leve, para obtener más información.
El umbral leve debe ser inferior al umbral. Esta función está desactivada de forma predeterminada. Cuando se especifica un umbral leve, su trabajo de perfil devuelve un recuento más,`zScoreMildOutliersCount`. También `zScoreOutliersSample` puede incluir una muestra de valores atípicos leves en este caso.

Configuración

- `threshold`— El valor umbral que se utilizará al detectar valores atípicos. Este valor debe ser mayor o igual a 0.
- `mildThreshold`— El valor umbral leve que se utilizará al detectar valores atípicos. Este valor debe ser mayor o igual a 0 e inferior `threshold` a.
- `sampleSize`— El tamaño de la lista que incluye los valores atípicos en la columna. Se utiliza `"ALL"` para enumerar todos los valores.

Ejemplo

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

DISTRIBUCIÓN DE VALORES

Mide la distribución de los valores de la columna según los rangos de los valores. Un trabajo de perfil agrupa los valores de una columna numérica o una columna de cadenas (en función de la longitud de la cadena) en grupos por rangos numéricos y genera una lista de grupos. Los intervalos son consecutivos y el límite superior de un depósito es el límite inferior del siguiente depósito.

Configuración

`binNumber`— Número de compartimentos. Este valor debe ser superior a 0.

Ejemplo

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

```
    }
}
```

Z_SCORE_DISTRIBUTION

Mide la distribución de las puntuaciones z de los valores en una columna numérica. Un trabajo de perfil agrupa las puntuaciones z de los valores en grupos por rangos numéricos y genera una lista de grupos. Los rangos son consecutivos y el límite superior de un grupo es el límite inferior del siguiente grupo.

Configuración

`binNumber`— Número de compartimentos. Este valor debe ser superior a 0.

Ejemplo

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

EntityDetectorConfiguration sección para configurar la PII

En la `EntityDetectorConfiguration` sección de su estructura, puede configurar los tipos de entidades de su conjunto de datos que desea DataBrew detectar como información de identificación personal (PII) para un puesto de perfil.

EntityTypes

Puede configurar los tipos de entidades que desea DataBrew detectar como PII para su puesto de perfil. Si no `EntityDetectorConfiguration` está definida, la detección de entidades está deshabilitada. Se pueden detectar los siguientes tipos de entidades en su conjunto de datos:

- 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

También `USA_ALL` se admite el grupo de tipos de entidades, que incluye todos los tipos de entidades anteriores, excepto `PERSON_NAME` y`DATE`.

El tipo de `EntityTypes` es una matriz de cadenas.

AllowedStatistics

Configure las estadísticas que se pueden ejecutar en las columnas que contienen entidades detectadas. Si no `AllowedStatistics` está definido, no se calculará ninguna estadística en las columnas que contienen entidades detectadas. Consulte Estadísticas configurables a nivel de columna para obtener una lista de valores válidos para el `AllowedStatistics` parámetro.

El tipo de `AllowedStatistics` es una matriz de `AllowedStatistics` objetos.

Seguridad en AWS Glue DataBrew

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El https://aws.amazon.com/compliance/shared-responsibility-model/ describe estos conceptos como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube:AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube.AWS también le proporciona servicios que puede utilizar de forma segura. Third-partylos auditores comprueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS Glue DataBrew, consulte los AWS servicios en el programa Scope by Compliance y AWS los servicios en el programa .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Glue DataBrew. Los siguientes temas muestran cómo configurarlo DataBrew para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus DataBrew recursos.

Temas

Protección de datos en AWS Glue DataBrew

DataBrew ofrece varias funciones diseñadas para ayudar a proteger sus datos.

Temas

El modelo de responsabilidad compartida de AWS se aplica a la protección de datos en AWS Glue DataBrew. Como se describe en este modelo,AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre privacidad de datos y los . Para obtener más información sobre la protección de datos en Europa, consulte el Centro del Reglamento General de Protección de Datos (RGPD).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management(IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.

- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja DataBrew o Servicios de AWS utiliza la consola, la API o los SDK.AWS CLIAWS Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

DataBrew admite el cifrado de datos en reposo para DataBrew proyectos y trabajos. Los proyectos y los trabajos pueden leer datos cifrados, y los trabajos pueden escribir datos cifrados llamando a AWS Key Management Service(AWS KMS) para generar claves y descifrar los datos. También puede usar las claves de KMS para cifrar los registros de trabajos generados por DataBrew los trabajos. Puede especificar las claves de cifrado mediante la DataBrew consola o la DataBrew API.

> **Important**
>
> AWS Glue DataBrew solo admite claves AWS KMS simétricas. Para obtener más información, consulte las claves AWS KMS en la Guía para AWS Key Management Service desarrolladores.

Al crear trabajos DataBrew con el cifrado activado, puede usar la DataBrew consola para especificar las claves de cifrado del S3-managed lado del servidor (SSE-S3) o las claves KMS almacenadas en AWS KMS(SSE-KMS) para cifrar los datos en reposo.

> **Important**
>
> Cuando utiliza un conjunto de datos de Amazon Redshift, los objetos descargados en el directorio temporal proporcionado se cifran con. SSE-S3

Cifrar los datos escritos por Jobs DataBrew

DataBrew los trabajos pueden escribir en destinos cifrados de Amazon S3 y en Amazon CloudWatch Logs cifrados.

Temas

¿Está configurando DataBrew el uso del cifrado?

Siga este procedimiento para configurar su DataBrew entorno para que utilice el cifrado.

Para configurar su DataBrew entorno para que utilice el cifrado

1. Cree o actualice sus claves de AWS KMS para conceder AWS KMS permisos a las funciones AWS Identity and Access Management(de IAM) que se transfieren a los DataBrew trabajos. Estas funciones de IAM se utilizan para cifrar los CloudWatch registros y los objetivos de Amazon S3. Para obtener más información, consulte Cifrar datos de registro en CloudWatch registros mediante AWS KMS la Guía del usuario de Amazon CloudWatch Logs.

   En el siguiente ejemplo, *"role1""role2"*, y *"role3"* son funciones de IAM que se transfieren a DataBrew los trabajos. Esta declaración de política describe una política de claves de KMS que permite a las funciones de IAM enumeradas cifrar y descifrar con esta clave de 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": "*"
    }
```

La `Service` declaración, que se muestra como`"Service": "logs.region.amazonaws.com"`, es obligatoria si se utiliza la clave para cifrar los registros. CloudWatch

2. Asegúrese de que la AWS KMS clave esté configurada en `ENABLED` antes de usarla.

Para obtener más información sobre cómo especificar los permisos mediante políticas AWS KMS clave, consulte Uso de políticas clave en AWS KMS.

Crear una ruta a AWS KMS para trabajos de VPC

Puede conectarse directamente a AWS KMS a través de un punto de enlace privado en su nube virtual privada (VPC) en lugar de conectarse a través de Internet. Cuando utiliza un punto final de VPC, la comunicación entre su VPC y la VPC AWS KMS se lleva a cabo íntegramente dentro de la red.AWS

Puede crear un punto final de AWS KMS VPC dentro de una VPC. Sin este paso, sus DataBrew trabajos podrían fallar con un. `kms timeout` Para obtener instrucciones detalladas, consulte Conexión a AWS KMS través de un punto final de VPC en la Guía para AWS Key Management Service desarrolladores.

Siguiendo estas instrucciones, en la consola de VPC, asegúrese de hacer lo siguiente:

- Selecciona Habilitar nombre DNS privado.
- En Grupo de seguridad, elija el grupo de seguridad (incluida una regla de autorreferencia) que utilizará para su DataBrew trabajo de acceso a la conectividad de bases de datos de Java (JDBC).

Cuando ejecute un DataBrew trabajo que acceda a los almacenes de datos de JDBC, DataBrew debe tener una ruta al punto final.AWS KMS Puede proporcionar a la ruta una puerta de enlace de traducción de direcciones de red (NAT) o un punto final de AWS KMS VPC. Para crear una gateway NAT, consulte Gateways NAT en la Guía del usuario de Amazon VPC.

Configurar el cifrado con AWS Claves de KMS

Cuando habilita el cifrado en un trabajo, se aplica tanto a Amazon S3 como a CloudWatch. La función de IAM que se transfiera debe tener los siguientes AWS KMS permisos.

Para obtener más información, consulte los siguientes temas en la Guía del usuario de Amazon Simple Storage Service:

- Para obtener más información`SSE-S3`, consulte Protección de datos mediante el Server-Side cifrado con claves de S3-Managed cifrado de Amazon (SSE-S3).
- Para obtener más información`SSE-KMS`, consulte Protección de datos mediante el Server-Side cifrado con claves AWS administradas por KMS (). SSE-KMS

Cifrado en tránsito

AWS proporciona cifrado Secure Sockets Layer (SSL) para los datos en movimiento.

DataBrew Incluye soporte para fuentes de datos JDBC.AWS Glue Al conectarse a fuentes de datos JDBC, DataBrew utiliza la configuración de su AWS Glue conexión, incluida la opción Requerir conexión SSL. Para obtener más información, consulte Propiedades de AWS Glue conexión AWS Glue en la Guía para AWS Glue desarrolladores.

AWS KMS proporciona el cifrado «traiga su propia clave» y el cifrado del lado del servidor para el procesamiento de DataBrew extracción, transformación y carga (ETL) y para el.AWS Glue Data Catalog

Administración de claves

Puede utilizar la IAM DataBrew para definir los usuarios, los AWS recursos, los grupos, las funciones y políticas detalladas en relación con el acceso, la denegación y mucho más.

Puede definir el acceso a los metadatos mediante políticas basadas en los recursos y en la identidad, en función de las necesidades de su organización. Resource-based las políticas enumeran los principales a los que se les permite o deniega el acceso a sus recursos, lo que le permite configurar

políticas como el acceso entre cuentas. Las políticas de identidades se asocian a los usuarios, grupos y roles dentro de IAM, de manera específica.

DataBrew permite crear su propio cifrado AWS KMS key«traiga su propia clave». DataBrew también proporciona cifrado del lado del servidor mediante claves KMS de AWS KMS for DataBrew jobs.

Identificación y manejo de la información de identificación personal (PII)

Al crear funciones analíticas o modelos de aprendizaje automático, se necesitan medidas de seguridad para evitar la exposición de los datos de información de identificación personal (PII). La PII es información personal que se puede utilizar para identificar a una persona, como una dirección, un número de cuenta bancaria o un número de teléfono. Por ejemplo, cuando los analistas y científicos de datos utilizan conjuntos de datos para descubrir información demográfica general, no deberían tener acceso a la PII de personas específicas.

DataBrew proporciona mecanismos de enmascaramiento de datos para ocultar los datos de PII durante el proceso de preparación de los datos. Según las necesidades de su organización, existen diferentes mecanismos de redacción de datos de PII disponibles. Puede ocultar los datos de PII para que los usuarios no puedan revertirlos, o puede hacer que la ofuscación sea reversible.

Identificar y enmascarar los datos de PII DataBrew implica crear un conjunto de transformaciones que los clientes puedan utilizar para redactar los datos de PII. Parte de este proceso consiste en proporcionar estadísticas y detección de datos de PII en el panel de información general del perfil de datos de la consola. DataBrew

Puede utilizar las siguientes técnicas de enmascaramiento de datos:

- Sustitución: sustituya los datos de PII por otros valores que parezcan auténticos.
- Mezcla: mezcla el valor de la misma columna en filas diferentes.
- Cifrado determinista: aplique algoritmos de cifrado determinista a los valores de las columnas. El cifrado determinista siempre produce el mismo texto cifrado para un valor.
- Cifrado probabilístico: aplique algoritmos de cifrado probabilístico a los valores de las columnas. El cifrado probabilístico produce un texto cifrado diferente cada vez que se aplica.
- Descifrado: descifra las columnas en función de las claves de cifrado.
- Anulación o eliminación: sustituya un campo concreto por un valor nulo o elimine la columna.
- Enmascaramiento: utilice la codificación de caracteres o oculte determinadas partes de las columnas.
- Algoritmos hash: aplique funciones hash a los valores de las columnas.

Para obtener más información sobre el uso de las transformaciones, consulta los pasos de la receta de información de identificación personal (PII). Para obtener más información sobre el uso de los trabajos de perfil para detectar la PII, incluida una lista de los tipos de entidades que se pueden detectar, consulte la EntityDetectorConfiguration sección sobre la configuración de la PII en Cómo crear una configuración de trabajo de perfil mediante programación.

DataBrew dependencia de otros AWS servicios

Para trabajar con la DataBrew consola, necesitas un conjunto mínimo de permisos para trabajar con los DataBrew recursos de tu AWS cuenta. Además de estos DataBrew permisos, la consola requiere los permisos de los siguientes servicios:

- CloudWatch Registra los permisos para mostrar los registros.
- Permisos de IAM para enumerar y transferir funciones.
- Permisos de Amazon EC2 para enumerar VPC, subredes, grupos de seguridad, instancias y otros objetos. DataBrew utiliza estos permisos para configurar elementos de Amazon EC2, como las VPC, al ejecutar tareas. DataBrew
- Permisos de Amazon S3 para enumerar buckets y objetos.
- AWS Glue permisos para leer objetos AWS Glue del esquema, como bases de datos, particiones, tablas y conexiones.
- AWS Lake Formation permisos para trabajar con lagos de datos de Lake Formation.

Gestión de identidad y acceso para AWS Glue DataBrew

AWS Identity and Access Management(IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos. DataBrew La IAM es una Servicio de AWS opción que puede utilizar sin coste adicional.

Temas

- Autenticación con identidades
- Administración del acceso con políticas
- AWS Glue DataBrew and AWS Lake Formation
- Cómo AWS Glue DataBrew funciona con IAM

- Identity-based ejemplos de políticas para AWS Glue DataBrew
- AWS políticas gestionadas para AWS Glue DataBrew
- Solución de problemas de identidad y acceso en AWS Glue DataBrew

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center(IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en la Cuenta de AWS en la Guía del usuario de AWS Sign-In.

Para el acceso programático,AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Al crear una Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz, que tiene acceso completo a todos los Servicios de AWS recursos. Se recomiendaencarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Usuarios y grupos

Un usuario de IAM es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un Rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso.AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Identity-based políticas

Identity-based las políticas son documentos de política de permisos de JSON que se adjuntan a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Identity-based las políticas pueden ser políticas integradas (integradas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte Selección entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Resource-based políticas

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe especificar una entidad principal en una política basada en recursos.

Resource-based las políticas son políticas en línea que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

DataBrew no admite políticas basadas en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL.AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

DataBrew no admite las ACL.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations.
- Políticas de control de recursos (RCP): definen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte Políticas de control de recursos (RCP) en la Guía del usuario de AWS Organizations.

- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

AWS Glue DataBrew and AWS Lake Formation

AWS Glue DataBrew admite AWS Lake Formation permisos para AWS Glue Data Catalog tablas. Cuando un conjunto de datos usa una AWS Glue Data Catalog tabla que está registrada en Lake Formation, la función de IAM proporcionada a los proyectos o trabajos debe tener los permisos DESCRIBE y SELECT Lake Formation en la tabla.

AWS Glue DataBrew permite escribir en AWS Glue Data Catalog tablas basándose en AWS Lake Formation. Cuando un DataBrew trabajo usa un catálogo de datos registrado en Lake Formation, la función de IAM proporcionada a los trabajos debe tener los permisos INSERT, ALTER y DELETE de Lake Formation para las tablas involucradas. El rol de IAM debe tener `glue:UpdateTable` permisos y también permisos para la ubicación de datos asociada a la tabla del catálogo de datos.

Cómo AWS Glue DataBrew funciona con IAM

Antes de utilizar IAM para gestionar el acceso DataBrew, debe comprender las funciones de IAM disponibles para su uso. DataBrew Para obtener una visión general de cómo funcionan con IAM DataBrew y otros AWS servicios, consulte AWS Servicios que funcionan con IAM en la Guía del usuario de IAM.

Temas

- DataBrew políticas basadas en la identidad
- Resource-based políticas en DataBrew
- DataBrew Funciones de IAM

DataBrew políticas basadas en la identidad

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados y también las condiciones en las que se permiten o deniegan las acciones. DataBrew admite acciones, claves de condiciones y recursos específicos. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte Referencia de los elementos de las políticas JSON de IAM en la Guía del usuario de IAM.

Acciones

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, una política de AWS JSON puede especificar qué director puede realizar acciones, en qué recursos y en qué condiciones.

El elemento Acción de una política de JSON describe las acciones a las que se puede permitir o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones políticas DataBrew utilizan el siguiente prefijo antes de la acción:`databrew:`. Por ejemplo, para conceder a alguien permiso para ejecutar una instancia de Amazon EC2 con la operación `RunInstances` de la API de Amazon EC2, debe incluir la acción `ec2:RunInstances` en la política. Las declaraciones de política deben incluir un `NotAction` elemento `Action` o. DataBrew define su propio conjunto de acciones que describen las tareas que puede realizar con él.

Para especificar varias acciones de en una única instrucción, sepárelas con comas del siguiente modo.

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

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción.

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

Para ver una lista de DataBrew acciones, consulte las acciones definidas por AWS Glue DataBrew en la Guía del usuario de IAM.

Recursos

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Las siguientes son las DataBrew API que no admiten permisos a nivel de recursos:

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

El recurso del DataBrew conjunto de datos tiene el siguiente nombre de recurso de Amazon (ARN).

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

Para obtener más información sobre el formato de los ARN, consulte Nombres de recursos de Amazon (ARN) y espacios de nombres de AWS servicio.

Por ejemplo, para especificar la instancia `i-1234567890abcdef0` en su instrucción, utilice el siguiente ARN.

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

Para especificar todas las instancias que pertenecen a una cuenta específica, utilice el carácter comodín (*).

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

No puede realizar algunas DataBrew acciones, como las de creación de recursos, en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

Para ver una lista de los tipos de DataBrew recursos y sus ARN, consulte los recursos definidos por AWS Glue DataBrew en la Guía del usuario de IAM. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS Glue DataBrew.

Claves de condición

DataBrew no proporciona ninguna clave de condición específica del servicio, pero sí admite el uso de algunas claves de condición globales. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Ejemplos

Para ver ejemplos de políticas DataBrew basadas en la identidad, consulte. Identity-based ejemplos de políticas para AWS Glue DataBrew

Resource-based políticas en DataBrew

DataBrew no admite políticas basadas en los recursos.

DataBrew Funciones de IAM

Un rol de IAM es una entidad de tu AWS cuenta que tiene permisos específicos.

Usar credenciales temporales con DataBrew

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Para obtener credenciales de seguridad temporales, puede llamar a operaciones de AWS STS API como AssumeRoleo GetFederationToken.

DataBrew admite el uso de credenciales temporales.

Service-linked roles

Service-linked los roles permiten a los AWS servicios acceder a los recursos de otros servicios para completar una acción en su nombre. Service-linked los roles aparecen en su cuenta de IAM y son propiedad del servicio. Un administrador puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Elegir un rol de IAM en DataBrew

Al crear un recurso de conjunto de datos en DataBrew, eliges un rol de IAM para permitir el DataBrew acceso en tu nombre. Si ha creado anteriormente un rol de servicio o un rol vinculado a un servicio, le DataBrew proporciona una lista de roles entre los que puede elegir. Asegúrese de elegir un rol que permita el acceso de lectura a un bucket o AWS Glue Data Catalog recurso de Amazon S3, según corresponda.

Identity-based ejemplos de políticas para AWS Glue DataBrew

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de DataBrew . Tampoco pueden realizar tareas con las AWS API Consola de administración de AWSAWS CLI, o. Un administrador debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos que necesiten esos permisos.

Para obtener información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Temas

- Prácticas recomendadas relativas a políticas
- Uso de la consola DataBrew
- Cómo permitir a los usuarios que vean sus propios permisos
- Administrar los DataBrew recursos en función de las etiquetas

Prácticas recomendadas relativas a políticas

Identity-based las políticas determinan si alguien puede crear DataBrew recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su.Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola DataBrew

Para acceder a la AWS Glue DataBrew consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los DataBrew recursos de su AWS cuenta. Si creas una política basada en la identidad que sea más restrictiva que los permisos mínimos requeridos, la consola no funcionará según lo previsto para los usuarios o roles con esa política.

Para garantizar que los usuarios y los roles puedan usar la DataBrew consola, adjunte también la siguiente política AWS administrada a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM.

```
AWSDataBrewConsoleAccess
```

No es necesario conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la DataBrew API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Cómo permitir a los usuarios que vean sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o 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": "*"
        }
    ]
}
```

Administrar los DataBrew recursos en función de las etiquetas

Puede usar las condiciones de su política basada en la identidad para administrar DataBrew los recursos en función de las etiquetas, por ejemplo, para eliminar, actualizar o describir los recursos. El siguiente ejemplo muestra una política que deniega la eliminación de un proyecto. Sin embargo, la eliminación solo se deniega si la etiqueta Owner del proyecto tiene el valor admin. Esta política también otorga los permisos necesarios para denegar esta acción en la consola.

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"}
            }
         }
      ]
}
```

También puede adjuntar esta política al usuario de en su cuenta. Si un usuario llamado richard-roe intenta eliminar un DataBrew proyecto, el recurso no debe tener la etiqueta owner=admin ni owner=admin. De lo contrario, se deniega al usuario el permiso para eliminar el proyecto. La clave de la etiqueta de condición Owner coincide con Owner y owner porque en los nombres de las claves de condición no se distingue entre mayúsculas y minúsculas. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

> **Note**
>
> ListDatasets, ListJobs, ListProjects ListRecipes ListRulesets, y ListSchedules no admiten el control de acceso basado en etiquetas.

AWS políticas gestionadas para AWS Glue DataBrew

Para añadir permisos a usuarios, grupos y roles, es más fácil usar políticas AWS administradas que escribirlas usted mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que proporcionen a su equipo solo los permisos necesarios. Para empezar rápidamente, puedes usar nuestras políticas AWS gestionadas. Estas políticas cubren casos de uso comunes y están disponibles en tu AWS cuenta. Para obtener más información sobre las políticas AWS administradas, consulte las políticas AWS administradas en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas. No puede cambiar los permisos en las políticas AWS gestionadas. En ocasiones, los servicios añaden permisos adicionales a una política AWS gestionada para admitir nuevas funciones. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Lo más probable es que los servicios actualicen una política AWS administrada cuando se lanza una nueva función o cuando hay nuevas operaciones disponibles. Los servicios no eliminan los permisos de una política AWS administrada, por lo que las actualizaciones de la política no afectarán a los permisos existentes.

Además,AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política ReadOnlyAccessAWS gestionada proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función,AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

DataBrew actualizaciones de AWS políticas administradas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas DataBrew desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial del DataBrew documento. La política gestionada se encuentra en la consola de AWS IAM en AwsGlueDataBrewFullAccessPolicy.

Cambio	Descripción	Fecha
AWSGlueDataBrewServiceRole— Se agregó el permiso de lectura para AWS Glue.	Esta actualización añade`glue:GetCustomEntityType` . Este permiso es necesario para ejecutar trabajos AWS Glue DataBrew de perfil si PII-identification está activado.	20 de marzo de 2024
AWSGlueDataBrewServiceRole- Se agregó el permiso de lectura para AWS Glue.	Esta actualización añade`glue:BatchGetCustomEntityTypes` . Este permiso es necesario para ejecutar trabajos AWS Glue DataBrew de perfil si PII-identification está activado.	9 de mayo de 2022
AwsGlueDataBrewFullAccessPolicy- GetLifecycleConfiguration Se han añadido permisos de lectura	Esta actualización se suma `redshift-data:DescribeStatement` a la compatibilidad con la	4 de febrero de 2022

Cambio	Descripción	Fecha
para Amazon Redshift-Data DescribeStatements y Amazon S3.	validación de su SQL al crear un Redshift-based conjunto de datos de Amazon. También sirve `s3:GetLifecycleConfiguration` para evaluar si el prefijo de bucket de Amazon S3 que está proporcionando como directorio temporal tiene el ciclo de vida configurado o no. Además, este cambio reemplaza los permisos «databrew: *» por una lista explícita de permisos que incluye todas las API. DataBrew	

Cambio	Descripción	Fecha
AwsGlueDataBrewFullAccessPolicy- se agregaron los Read/write permisos para AWS Secrets Manager.	Esta actualización añade `secretsmanager:CreateSecret` y, `secretsmanager:GetSecretValue` para un secreto denominado`databrew!default` , un secreto predeterminado para su uso en las DataBrew transformaciones. Además, añade permisos CreateSecret para los secretos con el prefijo `AwsGlueDataBrew-` para crearlos desde la DataBrew consola. GenerateRandom, que se describe en la referencia de la AWS Key Management Service API, se utiliza para generar una cadena de bytes aleatoria que es criptográficamente segura.	18 de noviembre de 2021
AWSGlueDataBrewServiceRole- se agregaron los Read/write permisos para AWS Secrets Manager.	Esta actualización añade `secretsmanager:GetSecretValue` un secreto denominado`databrew!default` , un secreto predeterminado para su uso en las DataBrew transformaciones.	18 de noviembre de 2021

Cambio	Descripción	Fecha
AwsGlueDataBrewFullAccessPolicy- se agregaron los Read/write permisos para AWS Secrets Manager.	Esta actualización añade `secretsmanager:CreateSecret` y, `secretsmanager:GetSecretValue` para un secreto denominado`databrew!default`, un secreto predeterminado para su uso en las DataBrew transformaciones. Además, añade permisos CreateSecret para los secretos con el prefijo `AwsGlueDataBrew-` para crearlos desde la DataBrew consola. `kms:GenerateRandom` (`https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html`) se utiliza para generar una cadena de bytes aleatoria que es criptográficamente segura.	18 de noviembre de 2021
AWSGlueDataBrewServiceRole- se agregaron los Read/write permisos para AWS Secrets Manager.	Esta actualización añade `secretsmanager:GetSecretValue` un secreto denominado`databrew!default`, un secreto predeterminado para su uso en las DataBrew transformaciones.	18 de noviembre de 2021

Cambio	Descripción	Fecha
AwsGlueDataBrewFullAccessPolicy- Se agregaron los permisos de lectura para AWS Glue las bases de datos del AWS Glue catálogo y los permisos de creación para la tabla del catálogo.	Esta actualización añade permisos para enumerar las bases de datos del AWS Glue catálogo y crear nuevas tablas de catálogo en una base de datos existente como parte de la configuración de la salida de los DataBrew trabajos.	30 de junio de 2021
AwsGlueDataBrewFullAccessPolicy- Se agregaron Read/write permisos para la función AppFlow de conjunto de datos de Amazon.	Esta actualización añade permisos para leer los AppFlow flujos y las ejecucion es de flujos de Amazon existentes y para crear ejecuciones de flujos.	28 de abril de 2021
AwsGlueDataBrewFullAccessPolicy- Se agregaron permisos de lectura para los conjuntos de datos de bases de datos.	Esta actualización añade permisos para leer AWS Glue las conexiones existentes y crear nuevas AWS Glue conexiones para usarlas con DataBrew ellas. Además, para facilitar la experiencia de la consola a la hora de crear nuevas conexiones, permite publicar los recursos de Amazon VPC y los clústeres de Amazon Redshift. También permite enumerar los secretos, pero no leerlos.AWS Secrets Manager	30 de marzo de 2021

Cambio	Descripción	Fecha
DataBrew comenzó a rastrear los cambios	DataBrew comenzó a rastrear los cambios de sus políticas AWS gestionadas.	30 de marzo de 2021

Solución de problemas de identidad y acceso en AWS Glue DataBrew

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con un DataBrew IAM.

Temas

No estoy autorizado a realizar ninguna acción en DataBrew

Si Consola de administración de AWS le indica que no está autorizado a realizar una acción, póngase en contacto con su administrador para obtener ayuda. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

En el siguiente ejemplo, el error se produce cuando el usuario de `mateojackson` intenta utilizar la consola para ver detalles sobre un proyecto pero no tiene permisos `databrew:DescribeProject`.

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

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *`my-example-project`* mediante la acción `databrew:`*`GetProject`*.

No estoy autorizado a realizar lo siguiente: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a DataBrew.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en DataBrew. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta para acceder a mis DataBrew recursos

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si DataBrew es compatible con estas funciones, consulteCómo AWS Glue DataBrew funciona con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Inicio de sesión y supervisión DataBrew

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de DataBrew sus AWS soluciones. Debe recopilar los datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente una falla multipunto en caso de que se produzca.AWS proporciona varias herramientas para supervisar sus DataBrew recursos y responder a posibles incidentes:

CloudWatch Alarmas Amazon

Con CloudWatch las alarmas de Amazon, observas una única métrica durante un período de tiempo que especifiques. Si la métrica supera un umbral determinado, se envía una notificación a un tema o AWS Auto Scaling política de Amazon SNS. CloudWatch las alarmas no invocan acciones porque se encuentran en un estado determinado. En su lugar, el estado debe haber cambiado y debe mantenerse durante el número de periodos especificado.

AWS CloudTrail Registros

CloudTrail proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en DataBrew. Con la información recopilada CloudTrail, puede determinar el destinatario de la solicitud DataBrew, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Validación de conformidad para AWS Glue DataBrew

Third-party los auditores evalúan la seguridad y el cumplimiento AWS Glue DataBrew como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento

Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte AWS la documentación de seguridad.

Resiliencia en AWS Glue DataBrew

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad.AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

En este AWS Glue DataBrew caso, le sugerimos que configure sus trabajos para que utilicen uno o más reintentos. El número de reintentos de un trabajo se configura en la DataBrew consola, en la sección Configuración avanzada del trabajo.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura AWS global.

Seguridad de la infraestructura en AWS Glue DataBrew

Como parte de un servicio gestionado,AWS Glue DataBrew está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las llamadas a la API AWS publicadas para acceder a DataBrew través de la red. Los clientes deben ser compatibles con la seguridad de la capa de transporte (TLS) 1.0 o una versión posterior. Recomendamos TLS 1.2 o una versión posterior. Los clientes también deben admitir conjuntos de cifrado con perfecto secreto directo (PFS), como Ephemeral (DHE) o Elliptic Curve Ephemeral Diffie-

Hellman (ECDHE). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Temas

Utilización AWS Glue DataBrew con tu VPC

Si utiliza Amazon VPC para alojar sus AWS recursos, puede configurarlo para enrutar el tráfico AWS Glue DataBrew a través de su nube privada virtual (VPC) en función del servicio Amazon VPC. DataBrew para ello, aprovisiona primero una interfaz de red elástica en la subred que especifique. DataBrew a continuación, adjunta el grupo de seguridad que especifique a esa interfaz de red para controlar el acceso. El grupo de seguridad especificado debe tener reglas de entrada y salida autorreferenciales para todo el tráfico. Además, la VPC debe tener activados los nombres de host DNS y la resolución. Para obtener más información, consulte Configuración de una VPC para conectarse a almacenes de datos JDBC en la Guía para desarrolladores.AWS Glue

En el AWS Glue Data Catalog caso de los conjuntos de datos, la información de la VPC se configura al crear AWS Glue una conexión en el catálogo de datos. Para crear tablas del catálogo de datos para esta conexión, ejecute un rastreador desde la consola.AWS Glue Para obtener más información, consulte Rellenar el AWS Glue Data Catalog en la Guía para AWS Glue desarrolladores.

Para los conjuntos de datos de bases de datos, especifique la información de la VPC al crear la conexión desde DataBrew la consola.

Para usarla AWS Glue DataBrew con una subred de VPC sin NAT, debe tener un punto de enlace de VPC de puerta de enlace a Amazon S3 y un punto de enlace de VPC para la interfaz.AWS Glue Para obtener más información, consulte Creación de un punto de enlace y puntos de enlace de VPC de interfaz (AWS PrivateLink) en la documentación de Amazon VPC. La interfaz elástica aprovisionada por DataBrew no tiene una dirección IPv4 pública, por lo que no admite el uso de una VPC Internet Gateway.

Los puntos de enlace de la interfaz Amazon S3 no son compatibles en este momento. Si vas AWS Secrets Manager a almacenar tu secreto, necesitas una ruta a Secrets Manager. Si utiliza el cifrado, necesita una ruta a AWS Key Management Service(AWS KMS).

Utilización AWS Glue DataBrew con puntos finales de VPC

Si utiliza Amazon VPC para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y DataBrew aprovisionar un punto de enlace de la VPC. Con este punto de conexión de VPC, puede realizar llamadas a la DataBrew API.

No es necesario utilizar un punto de enlace de DataBrew VPC DataBrew con su VPC. Para obtener más información, consulte Utilización AWS Glue DataBrew con tu VPC.

Puede usarlo AWS Glue con puntos de enlace de VPC en todas AWS las regiones que admitan ambos puntos de enlace de AWS Glue VPC.

Para obtener más información, consulte estos temas en la Guía del usuario de Amazon VPC:

- ¿Qué es Amazon VPC?
- Creación de un punto de enlace de interfaz

Análisis de configuración y vulnerabilidad en AWS Glue DataBrew

La configuración y los controles de TI son una responsabilidad compartida entre usted AWS y usted, nuestro cliente. Para obtener más información, consulte el modelo de responsabilidad AWS compartida.

Supervisión AWS Glue DataBrew

La supervisión es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de AWS Glue DataBrewAWS las demás soluciones.AWS proporciona las siguientes herramientas de monitoreo para observar DataBrew, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puede CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de sus instancias de Amazon EC2 y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Events le permite configurar notificaciones automáticas para eventos específicos en DataBrew. Los eventos de DataBrew se envían a CloudWatch Events prácticamente en tiempo real. Puede configurar CloudWatch los eventos para que supervisen los eventos e invoquen objetivos en respuesta a los eventos que indiquen cambios en sus recursos compartidos. Los cambios en un recurso compartido activan eventos tanto para el propietario del recurso compartido como para las entidades principales a las que se ha concedido acceso al recurso compartido. Para obtener más información, consulta la Guía del usuario de Amazon CloudWatch Events.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde instancias de Amazon EC2 y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcancen ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- AWS CloudTrailcaptura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre. A continuación, entrega los archivos log al bucket de Amazon S3 que se especifique. Puedes identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.

Temas

Monitorización DataBrew con Amazon CloudWatch

Puede monitorizar el DataBrew uso CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

AWS Glue DataBrew informa de las siguientes métricas en el espacio de `AWS/DataBrew` nombres.

Métrica	Description (Descripción)
`SessionCount`	El número total de DataBrew sesiones en la cuenta del cliente Dimensiones válidas: LogGroupName Estadísticas válidas: suma Unidades: recuento

Automatizar DataBrew con eventos CloudWatch

Amazon CloudWatch Events le permite automatizar sus AWS servicios y responder automáticamente a los eventos del sistema, como los problemas de disponibilidad de las aplicaciones o los cambios de recursos. Los eventos de AWS los servicios se envían a CloudWatch Events prácticamente en tiempo real. Puede crear reglas sencillas para indicar qué eventos le resultan de interés, así como qué acciones automatizadas se van a realizar cuando un evento cumple una de las reglas. Entre las acciones que se pueden activar automáticamente se incluyen las siguientes:

- Invocar el comando de ejecución de Amazon EC2
- Desviar el evento a Amazon Kinesis Data Streams
- Activar una máquina de AWS Step Functions estados
- Notificar un tema de Amazon SNS o una cola de Amazon SQS

DataBrew informa de un evento a CloudWatch Events cada vez que cambia el estado de un recurso de tu AWS cuenta. Los eventos se emiten en la medida de lo posible.

Los siguientes son ejemplos de varios eventos, que muestran varios estados de un DataBrew trabajo: `SUCCEEDED`,`FAILED`,`TIMEOUT`, y`STOPPED`.

```
{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-09-07T18:57:21Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
    "severity": "INFO",
    "state": "SUCCEEDED",
    "jobRunId": "db_abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789",
    "message": "Job run succeeded"
  }
}

{
  "version": "0",
  "id": "abcdef01-1234-5678-9abc-def012345678",
  "detail-type": "DataBrew Job State Change",
  "source": "aws.databrew",
  "account": "123456789012",
  "time": "2017-09-07T06:02:03Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "jobName": "MyJob",
```

```
    "severity": "ERROR",
    "state": "FAILED",
    "jobRunId": "db_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
    "message": "AnalysisException: 'Path does not exist: s3://MyBucket/MyFile;'"
  }
}

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

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

Para obtener más información, consulta la Guía del usuario de Amazon CloudWatch Events.

Supervisión DataBrew con CloudWatch registros

Puede supervisar los DataBrew trabajos mediante CloudWatch los registros, que recopilan información detallada del subsistema de DataBrew trabajos y la ponen a disposición para su revisión. Estos registros pueden serle útiles si desea obtener información sobre los recursos que utilizan su perfil y sus trabajos de recetas, o para solucionar problemas. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch Logs.

Registrar las llamadas a la DataBrew API con AWS CloudTrail

DataBrew está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en DataBrew. CloudTrail captura todas las llamadas a la API DataBrew como eventos. Las llamadas capturadas incluyen llamadas desde la DataBrew consola y llamadas en código a las operaciones de la DataBrew API. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para DataBrew. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud a la que se ha hecho DataBrew. También puede identificar la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario.

DataBrew Información en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crear la cuenta. Cuando se produce una actividad en DataBrew, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puedes ver, buscar y descargar los eventos recientes en tu AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos en la Guía del AWS CloudTrail usuario.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta DataBrew, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS

servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte lo indicado en la Guía del usuario de AWS CloudTrail:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas DataBrew las acciones se registran CloudTrail y se documentan en la referencia de la API. Por ejemplo, las llamadas a `UpdateRecipe` y `StartJobRun` las acciones generan entradas en los archivos de CloudTrail registro. `CreateDataset`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail .

Descripción de las entradas de los archivos de DataBrew registro

De nuevo, una CloudTrail ruta es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la `CreateProfileJob` operación.

```
{
    "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"
}
```

Utilización AWS Notificaciones de usuario con AWS Glue Databrew

Puede usar las notificaciones AWS de usuario para configurar canales de entrega para recibir notificaciones sobre los eventos de AWS Glue Databrew. Recibirá una notificación cuando un evento coincida con una regla que especifique. Puede recibir notificaciones de eventos a través de varios canales, como correo electrónico, notificaciones por chat de Amazon Q Developer en aplicaciones de chat o notificaciones push de AWS Console Mobile Application. También puede ver las notificaciones en el Centro de notificaciones de la consola.AWS Las notificaciones de usuario

admiten la agregación, lo que puede reducir la cantidad de notificaciones que recibe durante eventos específicos.

Referencia de pasos y funciones de la receta

En esta referencia, encontrarás descripciones de los pasos y funciones de la receta que puedes usar mediante programación, ya sea desde AWS CLI o mediante uno de los SDK.AWS En este DataBrew caso, un paso de receta es una acción que transforma los datos sin procesar en un formulario que está listo para ser utilizado por la canalización de datos. Una DataBrew función es un tipo especial de paso de receta que realiza un cálculo basado en parámetros.

Entre las categorías de transformaciones de la interfaz de usuario se incluyen las siguientes:

- Pasos básicos de la receta de columnas
  - Filtro
  - Columna
- Pasos de la receta de limpieza de datos
  - Formato
  - Limpio
  - Extract
- Pasos de la receta de calidad de los
  - Missing (Ausente)
  - Invalid (No válido)
  - Duplicados
  - Valores atípicos
- Pasos de la receta de información de identificación personal (PII)
  - Enmascarar la información personal
  - Sustituir la información personal
  - Cifra la información personal
  - Mezclar filas
- Pasos de la receta de estructura de columnas
  - Split
  - Merge
  - Crear
- Pasos de la receta de formato de columnas

  - Precisión decimal
  - Separador de miles
  - Abrevia números
- Pasos de la receta de estructura de datos
  - Nest-Unnest
  - Pivot
  - Group
  - Join
  - Unión
- Pasos de la receta de ciencia de datos
  - Texto
  - Escalado
  - Correspondencia
  - Codificación
- Funciones
  - Funciones matemáticas
  - Funciones de agregación
  - Funciones de texto
  - Funciones de fecha y hora
  - Funciones de ventana
  - Funciones web
  - Otras funciones

Para obtener más información sobre cómo se utilizan estos pasos y funciones en una receta (incluido el uso de expresiones condicionales), consulteDefinir la estructura de una receta.

En las siguientes secciones se describen los pasos y las funciones de la receta, organizados según su función.

Temas

- Pasos básicos de la receta en columnas
- Pasos de la receta de limpieza de datos

Pasos básicos de la receta en columnas

Utilice estas acciones básicas de receta de columnas para realizar transformaciones sencillas en sus datos.

Temas

- RENAME
- SORT
- A _BOOLEAN_COLUMN
- A UNA COLUMNA DOBLE
- A _NÚMERO_COLUMNA
- TO_STRING_COLUMN

CAMBIAR_TIPO_DE_DATOS

Cambia el tipo de datos de una columna existente.

Si el valor de una columna no se puede convertir al nuevo tipo, se sustituirá por NULL. Esto puede suceder cuando una columna de cadena se convierte en una columna de enteros. Por ejemplo, la cadena «123" pasará a ser el entero 123, pero la cadena «ABC» no puede convertirse en un número, por lo que se sustituirá por un valor NULO.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— Nuevo tipo de columna. Los siguientes tipos de datos son compatibles:
  - byte: números enteros con signo de 1 byte. El rango de números va de -128 a 127.
  - corto: números enteros con signo de 2 bytes. El rango de números va de -32768 a 32767.
  - int: números enteros con signo de 4 bytes. El rango de números va de -2147483648 a 2147483647.
  - largo: números enteros con signo de 8 bytes. El rango de números va de -9223372036854775808 a 9223372036854775807.
  - float: números de coma flotante de precisión simple de 4 bytes.
  - doble: números de coma flotante de precisión doble de 8 bytes.
  - decimal: números decimales firmados con hasta 38 dígitos en total y 18 dígitos después de la coma decimal.
  - cadena: valores de cadenas de caracteres.
  - booleano: el tipo booleano tiene uno de dos valores posibles: `verdadero` y `falso` o `sí` y `no`.
  - marca de tiempo: valores que comprenden los campos año, mes, día, hora, minuto y segundo.
  - fecha: valores que comprenden los campos año, mes y día.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
 }
```

DELETE

Elimina una columna del conjunto de datos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

DUPLICAR

Crea una nueva columna con un nombre diferente, pero con todos los mismos datos. Tanto la columna antigua como la nueva se conservan en el conjunto de datos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `targetColumn`— Un nombre para la columna duplicada.

Example Ejemplo

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

JSON_TO_STRUCTS

Convierte una cadena JSON en estructuras de tipo estático. Durante la conversión, detecta el esquema de cada objeto JSON y los fusiona para obtener el esquema más genérico que represente toda la cadena JSON. El parámetro «UnnestLevel» especifica cuántos niveles de objetos JSON se deben convertir en estructuras.

Parameters

- `sourceColumns`— Una lista de columnas de origen.
- `regexColumnSelector` –Una expresión regular para seleccionar las columnas.
- `removeSourceColumn`— Un valor booleano. Si es `true` así, elimina la columna de origen; de lo contrario, consérvala.
- `unnestLevel`— El número de niveles que se van a deshacer.
- `conditionExpressions`— Expresiones de condición.

Example Ejemplo

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

```
            "sourceColumns": "[\"address\"]",
            "removeSourceColumn": "true",
            "unnestLevel": "2"
        }
    }
}
```

MOVE_AFTER

Mueve una columna a la posición inmediatamente posterior a otra columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`— El nombre de otra columna. La columna especificada por se `sourceColumn` moverá inmediatamente después de la columna especificada por`targetColumn`.

Example Ejemplo

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

MOVE_BEFORE

Mueve una columna a la posición inmediatamente anterior a otra columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`— El nombre de otra columna. La columna especificada por se `sourceColumn` moverá inmediatamente después de la columna especificada por`targetColumn`.

Example Ejemplo

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

MOVE_TO_END

Mueve una columna a la posición final (última columna) del conjunto de datos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

MOVE_TO_INDEX

Mueve una columna a una posición especificada por un número.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetIndex`— La nueva posición de la columna. Las posiciones comienzan por 0, por ejemplo, 1 se refieren a la segunda columna, 2 a la tercera columna, etc.

Example Ejemplo

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

MOVER_TO_EMPEZAR

Mueve una columna a la posición inicial (primera columna) del conjunto de datos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

RENAME

Crea una nueva columna con un nombre diferente, pero con todos los mismos datos. A continuación, la columna anterior se elimina del conjunto de datos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `targetColumn`— Un nombre nuevo para la columna.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

Ordena los datos de una o más columnas de un conjunto de datos en orden ascendente, descendente o personalizado.

Parameters

- `expressions`— Una cadena que contiene una o más JSON-encoded cadenas que representan expresiones de ordenación.
  - `sourceColumn`— Una cadena que contiene el nombre de una columna existente.
  - `ordering`— El orden puede ser ASCENDENTE o DESCENDENTE.
  - `nullsOrdering`— El orden de los valores nulos puede ser NULLS_TOP o NULLS_BOTTOM para colocar los valores nulos o faltantes al principio o al final de la columna.
  - `customOrder`— Una lista de cadenas que define un orden personalizado para la clasificación de las cadenas. De forma predeterminada, las cadenas se ordenan alfabéticamente.
  - `isCustomOrderCaseSensitive`: booleano. El valor predeterminado es `false`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SORT",
```

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

Example Ejemplo de orden de clasificación personalizado

En el ejemplo siguiente, la cadena de expresión CustomOrder tiene el formato de una lista de objetos. Cada objeto describe una expresión de ordenación para una columna.

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

A _BOOLEAN_COLUMN

Cambia el tipo de datos de una columna existente a BOOLEANO.

> **Note**
>
> Se recomienda utilizar la acción de receta CHANGE_DATA_TYPE en lugar de TO_BOOLEAN_COLUMN.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `columnDataType`— Un valor `boolean` que debe serlo.

Example Ejemplo

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

A UNA COLUMNA DOBLE

Cambia el tipo de datos de una columna existente a DOUBLE.

> **Note**
>
> Se recomienda utilizar la acción de receta CHANGE_DATA_TYPE en lugar de TO_DOUBLE_COLUMN.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— Un valor `number` que debe serlo.

Example Ejemplo

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

```
    }
}
```

A _NÚMERO_COLUMNA

Cambia el tipo de datos de una columna existente a NUMBER.

> **Note**
>
> Se recomienda utilizar la acción de receta CHANGE_DATA_TYPE en lugar de TO_NUMBER_COLUMN.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— Un valor `number` que debe serlo.

Example Ejemplo

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

TO_STRING_COLUMN

Cambia el tipo de datos de una columna existente a STRING.

> **Note**
>
> Se recomienda utilizar la acción de receta CHANGE_DATA_TYPE en lugar de TO_STRING_COLUMN.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— Un valor `string` que debe serlo.

Example Ejemplo

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

Pasos de la receta de limpieza de datos

Siga estos pasos de la receta de limpieza de datos para realizar transformaciones sencillas en los datos existentes.

Temas

- EXTRACT_PATTERN
- EXTRACT_VALUE
- REMOVE_COMBINED
- REPLACE_BETWEEN_DELIMITERS
- SUSTITUIR_ENTRE_POSICIONES
- REPLACE_TEXT

CAPITAL_CASE

Cambia cada cadena de una columna para poner en mayúscula cada palabra. En mayúscula, la primera letra de cada palabra se escribe en mayúscula y el resto de la palabra se transforma en minúscula. Un ejemplo es: El veloz zorro marrón saltó por encima de la valla.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

FORMAT_DATE

Devuelve una columna en la que una cadena de fecha se convierte en un valor formateado.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetDateFormat`— Uno de los siguientes formatos de fecha:

- mm/dd/yyyy
- mm-dd-yyyy
- dd month yyyy
- month yyyy
- dd month

Example Ejemplo

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

MINÚSCULAS/MINÚSCULAS

Cambia cada cadena de una columna a minúsculas, por ejemplo: el veloz zorro marrón saltó la cerca

Parameters

- sourceColumn: el nombre de una columna existente.

Example Ejemplo

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

MAYÚSCULAS_MAYÚSCULAS

Cambia cada cadena de una columna a mayúsculas, por ejemplo: THE QUICK BROWN FOX JUMPT OVER THE FENCE

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

SENTENCE_CASE

Cambia cada cadena de una columna a mayúsculas y minúsculas. En el caso de una oración, la primera letra de cada oración se escribe en mayúscula y el resto de la oración se transforma en minúscula. Un ejemplo es: El zorro pardo veloz. Saltó por encima. ¿La cerca

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

```
}
```

ADD_DOUBLE_QUOTES

Incluye los caracteres de una columna entre comillas dobles.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

ADD_PREFIX

Añade uno o más caracteres y los concatena como prefijo al principio de una columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— El carácter o los caracteres que se van a colocar al principio de los valores de la columna.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ADD_PREFIX",
        "Parameters": {
            "pattern": "aaa",
```

```
            "sourceColumn": "info_url"
        }
    }
}
```

AÑADIR_COMILLAS SIMPLES

Incluye los caracteres de una columna entre comillas simples.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

ADD_SUFFIX

Agrega un carácter más y los concatena como un sufijo al final de una columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— El carácter o los caracteres que se van a colocar al final de la columna.

Example Ejemplo

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

```
            "pattern": "bbb",
            "sourceColumn": "info_url"
        }
    }
}
```

EXTRACT_BETWEEN_DELIMITERS

Crea una nueva columna, basada en los delimitadores, a partir de los valores de una columna existente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `startPattern`— Una expresión regular que indica el carácter o los caracteres que comienzan los valores delimitados.
- `endPattern`— Una expresión regular que indica el carácter o los caracteres delimitadores que terminan los valores delimitados.

Example Ejemplo

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

EXTRACT_ENTRE_POSICIONES

Crea una nueva columna, basada en las posiciones de los caracteres, a partir de los valores de una columna existente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `startPosition`— La posición del personaje en la que se va a realizar la extracción.
- `endPosition`— La posición del personaje en la que se debe finalizar la extracción.

Example Ejemplo

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

EXTRACT_PATTERN

Crea una nueva columna, basada en una expresión regular, a partir de los valores de una columna existente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `pattern`— Una expresión regular que indica el carácter o los caracteres que se van a extraer y a partir de los cuales se va a crear la nueva columna.

Example Ejemplo

```
{
```

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

EXTRACT_VALUE

Crea una nueva columna con un valor extraído de una ruta especificada por el usuario. Si la columna de origen es del tipo Mapa, Matriz o Estructura, se debe separar cada campo de la ruta utilizando marcas inversas (por ejemplo, `nombre`).

Parameters

- `targetColumn`— El nombre de la columna de destino.
- `sourceColumn`— Nombre de la columna de origen de la que se va a extraer el valor.
- `path`— La ruta a la clave específica que el usuario quiere extraer. Si la columna de origen es del tipo Mapa, Matriz o Estructura, se debe separar cada campo de la ruta con marcas inversas (por ejemplo, `nombre`).

  Considere el siguiente ejemplo de información de usuario:

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

  Los siguientes son ejemplos de las rutas que proporcionaría, según el tipo de columna de origen:

- Si la columna de origen es del tipo mapa, la ruta para extraer el número de teléfono fijo es:

`user`.`phoneNumber`.`home`

- Si la columna de origen es del tipo Array, la ruta para extraer el segundo valor de «ciudadanía» es:

  `user`.`citizenship`[1]

- Si la columna de origen es del tipo struct, la ruta para extraer el código postal es:

  `user`.`address`.`zipcode`

Example Ejemplo

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

REMOVE_COMBINED

Elimina uno o más caracteres de una columna, según lo que especifique el usuario.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `collapseConsecutiveWhitespace`— Si`true`, reemplaza dos o más caracteres de espacio en blanco por exactamente un carácter de espacio en blanco.
- `removeAllPunctuation`— Si`true`, elimina todos los caracteres siguientes: `. ! , ?`
- `removeAllQuotes`— Si`true`, elimina todas las comillas simples y dobles.
- `removeAllWhitespace`— Si`true`, elimina todos los espacios en blanco.
- `customCharacters`— Uno o más personajes sobre los que se puede actuar.

- `customValue`— Un valor sobre el que se puede actuar.
- `removeCustomCharacters`— Si`true`, elimina todos los caracteres especificados por el `customCharacters` parámetro.
- `removeCustomValue`— Si`true`, elimina todos los caracteres especificados por el `customValue` parámetro.
- `punctuationally`— Si`true`, elimina los siguientes caracteres si aparecen al principio o al final del valor: `. ! , ?`
- `antidisestablishmentarianism`— Si`true`, elimina las comillas simples y dobles del principio y del final del valor.
- `removeLeadingAndTrailingWhitespace`— Si`true`, elimina todos los espacios en blanco del principio y del final del valor.
- `removeLetters`— Si`true`, elimina todos los caracteres alfabéticos en mayúscula y minúscula (de principio `A` a `Z` fin). a `z`
- `removeNumbers`— Si`true`, elimina todos los caracteres numéricos (de principio a fin). `0` `9`
- `removeSpecialCharacters`— Si`true`, elimina todos los caracteres siguientes: `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`

Example Ejemplos

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

```
        }
    }
}
```

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

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

```
            "sourceColumn": "info_url"
        }
    }
}
```

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

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

```
        }
    }
}
```

REPLACE_BETWEEN_DELIMITERS

Sustituye los caracteres entre dos delimitadores por texto especificado por el usuario.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `startPattern`— Carácter o caracteres o una expresión regular, que indican dónde va a empezar la sustitución.
- `endPattern`— Carácter o caracteres o una expresión regular, que indique dónde va a terminar la sustitución.
- `value`— El carácter o los caracteres sustitutivos que se van a sustituir.

Example Ejemplo

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

SUSTITUIR_ENTRE_POSICIONES

Sustituye los caracteres entre dos posiciones por texto especificado por el usuario.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `startPosition`— Un número que indica en qué posición de carácter de la cadena debe comenzar la sustitución.
- `endPosition`— Un número que indica en qué posición de caracteres de la cadena debe terminar la sustitución.
- `value`— El carácter o los caracteres de reemplazo que se van a sustituir.

Example Ejemplo

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

REPLACE_TEXT

Sustituye una secuencia de caracteres especificada por otra.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— Carácter o caracteres o una expresión regular, que indique qué caracteres deben sustituirse en la columna de origen.
- `value`— El carácter o los caracteres sustitutivos que se van a sustituir.

Example Ejemplos

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

```
            "pattern": "x",
            "sourceColumn": "first_name",
            "value": "a"
        }
    }
}
```

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

Pasos de la receta de calidad de datos

Siga estos pasos de la receta de calidad de los datos para rellenar los valores faltantes, eliminar los datos no válidos o eliminar los duplicados.

Temas

- FILTRO_DE_TIPO_DATOS AVANZADO
- ADVANCED_DATATYPE_FLAG
- DELETE_DUPLICATE_ROWS
- EXTRACT_ADVANCED_DATATYPE_DETAILS
- RELLÉN_CON_PROMEDIO
- RELLÉN_CON_PERSONALIZADO
- RELLÉN_CON_VACÍO
- RELLÉNALA CON_LAST_VALID
- FILL_WITH_MEDIAN
- FILL_WITH_MODE
- RELLÉNELO CON LO MÁS FRECUENTE
- RELLÉN_CON_NULL

- RELLÉN_CON_SUMA
- FLAG_DUPLICATE_ROWS
- FLAG_DUPLICATES_IN_COLUMN
- GET_ADVANCED_DATATYPE
- REMOVE_DUPLICATES
- REMOVE_INVALID
- REMOVE_MISSING
- SUSTITUIR_CON_PROMEDIO
- SUSTITUIR_CON_PERSONALIZADO
- SUSTITUIR_CON_VACÍO
- REEMPLACE_CON_LAST_VALID
- SUSTITUYE_CON_MEDIAN
- REPLACE_WITH_MODE
- REEMPLACE_CON_MOST_FREQUENT
- SUSTITUIR_CON_NULL
- REEMPLÁZALA POR UNA MEDIA VARIABLE
- REEMPLACE_CON_ROLLING_SUM
- SUSTITUIR_CON_SUMA

FILTRO_DE_TIPO_DATOS AVANZADO

Filtra la columna de origen actual en función de la detección avanzada de tipos de datos. Por ejemplo, si se identifica una columna que DataBrew contiene códigos postales, esta transformación puede filtrar la columna en función de la zona horaria. Los detalles que se pueden extraer dependen del patrón que se detecte, tal y como se describe en las notas siguientes.

Parameters

- `sourceColumn`— El nombre de una columna de origen de cadenas.
- `pattern`— El patrón que se va a extraer.
- `advancedDataType`— Puede ser uno de los siguientes: teléfono, código postal, fecha, hora, estado, tarjeta de crédito, URL, correo electrónico, número de seguro social o sexo.

- `filter values`— Lista de valores de cadena por los que el usuario quiere filtrar la columna.
- `strategy`— KEEP_ROWS o DISCARD_ROWS o CLEAR_FILTERS o CLEAR_OTHERS.
- `clearWithEmpty`— `true` Booleano o, para borrar filas con ellas en lugar de. `false` `empty` `null`

Notas

- Si la opción avanzada DataType es Teléfono, el patrón puede ser AREA_CODE, TIME_ZONE o COUNTRY_CODE.
- Si el nivel avanzado DataType es Zip Code, el patrón puede ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE o REGION.
- Si avanzado DataType es Fecha y hora, el patrón puede ser DAY, MONTH_NAME, WEEK, QUARTER o YEAR.
- Si avanzado DataType es Estado, el patrón puede ser TIME_ZONE.
- Si el valor avanzado DataType es Tarjeta de crédito, el patrón puede ser LONGITUD o RED.
- Si avanzado DataType es URL, el patrón puede ser PROTOCOLO, TLD o DOMINIO.

Example Ejemplo

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

ADVANCED_DATATYPE_FLAG

Crea una nueva columna indicadora basada en los valores de la columna de origen actual. Por ejemplo, si una columna de origen contiene códigos postales, esta transformación se puede utilizar para marcar valores como `true` o en `false` función de una zona horaria concreta. Los detalles

que se pueden extraer dependen del patrón que se detecte, tal y como se describe en las notas siguientes.

Parameters

- `sourceColumn`— El nombre de una columna de origen de cadenas.
- `pattern`— El patrón que se va a extraer.
- `targetColumn`— El nombre de la columna de destino.
- `advancedDataType`— Puede ser uno de los siguientes: teléfono, código postal, fecha, hora, estado, tarjeta de crédito, URL, correo electrónico, número de seguro social o sexo.
- `filter values`— Lista de valores de cadena por los que el usuario quiere filtrar la columna.
- `trueString`— El `true` valor de la columna de destino.
- `falseString`— El `false` valor de la columna de destino.

Notas

- Si avanzado DataType es Teléfono, el patrón puede ser AREA_CODE, TIME_ZONE o COUNTRY_CODE.
- Si el nivel avanzado DataType es Zip Code, el patrón puede ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE o REGION.
- Si avanzado DataType es Fecha y hora, el patrón puede ser DAY, MONTH_NAME, WEEK, QUARTER o YEAR.
- Si avanzado DataType es Estado, el patrón puede ser TIME_ZONE.
- Si el valor avanzado DataType es Tarjeta de crédito, el patrón puede ser LONGITUD o RED.
- Si avanzado DataType es URL, el patrón puede ser PROTOCOLO, TLD o DOMINIO.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ADVANCED_DATATYPE_FLAG",
        "Parameters": {
            "pattern": "AREA_CODE",
            "sourceColumn": "phoneColumn",
            "advancedDataType": "Phone",
```

```
            "filterValues": ['Ohio'],
            "targetColumn": "targetColumnName",
            "trueString": "trueValue",
            "falseString": "falseValue"
        }
    }
}
```

DELETE_DUPLICATE_ROWS

Elimina cualquier fila que coincida exactamente con una fila anterior del conjunto de datos. La aparición inicial no se elimina porque no coincide con una fila anterior.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRACT_ADVANCED_DATATYPE_DETAILS

Extrae los detalles del tipo de datos avanzado. Los detalles que puede extraer dependen del patrón que se detecte, tal y como se describe en las notas siguientes.

Parameters

- `sourceColumn`— El nombre de una columna de origen de cadenas.
- `pattern`— El patrón que se va a extraer.
- `targetColumn`— El nombre de la columna de destino.
- `advancedDataType`— Puede ser uno de los siguientes: teléfono, código postal, fecha, hora, estado, tarjeta de crédito, URL, correo electrónico, número de seguro social o sexo.

Notas

- Si la opción avanzada DataType es Teléfono, el patrón puede ser AREA_CODE, TIME_ZONE o COUNTRY_CODE.

- Si el nivel avanzado DataType es Zip Code, el patrón puede ser TIME_ZONE, COUNTRY, STATE, CITY, TYPE o REGION.
- Si avanzado DataType es Fecha y hora, el patrón puede ser DAY, MONTH_NAME, WEEK, QUARTER o YEAR.
- Si avanzado DataType es Estado, el patrón puede ser TIME_ZONE.
- Si el valor avanzado DataType es Tarjeta de crédito, el patrón puede ser LONGITUD o RED.
- Si avanzado DataType es URL, el patrón puede ser PROTOCOLO, TLD o DOMINIO.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

RELLÉN_CON_PROMEDIO

Devuelve una columna en la que faltan datos reemplazados por el promedio de todos los valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
```

```
        }
    }
}
```

RELLÉN_CON_PERSONALIZADO

Devuelve una columna en la que faltan datos reemplazados por un valor específico.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser `datenumber`,`boolean`,`unsupported`,`string`, `otimestamp`.
- `value`— El valor personalizado que se va a rellenar. El tipo de datos debe coincidir con el valor que elija`columnDataType`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

RELLÉN_CON_VACÍO

Devuelve una columna con datos faltantes reemplazados por una cadena vacía.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
}
```

RELLÉNALA CON_LAST_VALID

Devuelve una columna en la que faltan datos reemplazados por el valor válido más reciente de esa columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser `datenumber`,`boolean`,`unsupported`,`string`, `otimestamp`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

FILL_WITH_MEDIAN

Devuelve una columna en la que faltan datos reemplazados por la mediana de todos los valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MODE

Devuelve una columna en la que faltan datos reemplazados por el modo de todos los valores.

También puede especificar una lógica de desempate, en la que algunos valores son idénticos. Por ejemplo, considere los siguientes valores:

`1 2 2 3 3 4`

A `modeType` de `MINIMUM` hace `FILL_WITH_MODE` que devuelva 2 como valor de modo. Si `modeType` es `MAXIMUM` así, el modo es 3. Para `AVERAGE`, el modo es 2,5.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `modeType`: cómo resolver los valores de empate en los datos. Este valor debe ser `MINIMUMNONE`,`AVERAGE`, o`MAXIMUM`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
```

```
    }
}
```

RELLÉNELO CON LO MÁS FRECUENTE

Devuelve una columna en la que faltan datos reemplazados por el valor más frecuente.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
    }
}
```

RELLÉN_CON_NULL

Devuelve una columna con valores de datos sustituidos por nulos.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
```

```
    }
}
```

RELLÉN_CON_SUMA

Devuelve una columna en la que faltan datos reemplazados por la suma de todos los valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FLAG_DUPLICATE_ROWS

Devuelve una nueva columna con un valor específico en cada fila que indica si esa fila coincide exactamente con una fila anterior del conjunto de datos. Cuando se encuentran coincidencias, se marcan como duplicadas. La aparición inicial no está marcada porque no coincide con una fila anterior.

Parameters

- `trueString`: valor que se insertará si la fila coincide con una fila anterior.
- `falseString`: valor que se insertará si la fila es única.
- `targetColumn`: nombre de la nueva columna que se inserta en el conjunto de datos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

FLAG_DUPLICATES_IN_COLUMN

Devuelve una nueva columna con un valor especificado en cada fila que indica si el valor de la columna de origen de la fila coincide con un valor de una fila anterior de la columna de origen. Cuando se encuentran coincidencias, se marcan como duplicadas. La aparición inicial no está marcada porque no coincide con una fila anterior.

Parameters

- `sourceColumn`: nombre de la columna de origen.
- `targetColumn`: nombre de la columna de destino.
- `trueString`: cadena que se insertará en la columna de destino cuando el valor de una columna de origen duplique un valor anterior de esa columna.
- `falseString`: cadena que se insertará en la columna de destino cuando el valor de una columna de origen sea distinto de un valor anterior de esa columna.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
```

```
}
```

GET_ADVANCED_DATATYPE

Dada una columna de cadena, identifica el tipo de datos avanzados de la columna, si lo hay.

Parameters

- `columnName`— El nombre de la columna de cadena.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

REMOVE_DUPLICATES

Elimina una fila completa si se encuentra un valor duplicado en una columna de origen seleccionada.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

```
}
```

REMOVE_INVALID

Elimina una fila completa si se encuentra un valor no válido en una columna de esa fila.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
}
```

REMOVE_MISSING

Devuelve solo las filas en las que no faltan datos a una columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.

Example Ejemplo

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

SUSTITUIR_CON_PROMEDIO

Sustituye cada valor no válido de una columna por el promedio de todos los demás valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "age"
        }
    }
}
```

SUSTITUIR_CON_PERSONALIZADO

Reemplace las entidades detectadas por un valor personalizado.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `sourceColumns`— Una lista de nombres de columnas existentes.
- `columnDataType`— El tipo de datos de la columna.

- `value`— El valor personalizado que se utilizará para reemplazar los valores no válidos.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

> **Note**
>
> Utilice una `sourceColumn` de las dos`sourceColumns`, pero no ambas.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

SUSTITUIR_CON_VACÍO

Sustituye cada valor no válido de una columna por un valor vacío.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de

teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

REEMPLACE_CON_LAST_VALID

Sustituye cada valor no válido de una columna por el último valor válido.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
```

```
    }
}
```

SUSTITUYE_CON_MEDIAN

Sustituye cada valor no válido de una columna por la mediana de todos los demás valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

REPLACE_WITH_MODE

Sustituye cada valor no válido de una columna por el modo de todos los demás valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.
- `modeType`: cómo resolver los valores de empate en los datos. Este valor debe ser `MINIMUMNONE`,`AVERAGE`, o`MAXIMUM`.

Example Ejemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
    }
}
```

REEMPLACE_CON_MOST_FREQUENT

Sustituye cada valor no válido de una columna por el valor de columna más frecuente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "wind_direction"
        }
    }
}
```

SUSTITUIR_CON_NULL

Sustituye cada valor no válido de una columna por un valor nulo.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna.
- `advancedDataType`— Tipos de datos especiales que se detectan DataBrew en una columna que contiene ese tipo de datos`string`. Entre los tipos que DataBrew se pueden detectar en una `string` columna se incluyen el número de seguro social, el correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP DateTime, la divisa ZipCode, el país, la región, el estado y la ciudad.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

REEMPLÁZALA POR UNA MEDIA VARIABLE

Sustituye cada valor de una columna por el promedio móvil de una «ventana» anterior de filas.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.
- `period`- — El tamaño de la ventana. Por ejemplo, si `period` es 10, la media móvil se calcula utilizando las 10 filas anteriores.

Example Ejemplo

```
{
    "RecipeStep": {
```

```
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REEMPLACE_CON_ROLLING_SUM

Sustituye cada valor de una columna por la suma acumulada de una «ventana» anterior de filas.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.
- `period`- — El tamaño de la ventana. Por ejemplo, si `period` es 10, la suma acumulada se calcula utilizando las 10 filas anteriores.

Example Ejemplo

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

SUSTITUIR_CON_SUMA

Sustituye cada valor no válido de una columna por la suma de todos los demás valores.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `columnDataType`— El tipo de datos de la columna. Este tipo debe ser`number`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

Pasos de la receta de información de identificación personal (PII)

Siga estos pasos de la receta para realizar transformaciones en la información de identificación personal (PII) de un conjunto de datos.

> **Note**
>
> Además de los pasos de receta de esta sección, hay pasos de DataBrew receta que no están diseñados específicamente para la PII y que puede utilizar para gestionar la PII. Un ejemplo es DELETE un paso básico de receta de columnas que elimina una columna.

Temas

- CRYPTOGRAPHIC_HASH
- DESCIFRAR
- DETERMINISTIC_DECRYPT
- DETERMINISTIC_ENCRYPT
- ENCRIPTAR
- MASK_CUSTOM

- MASK_DATE
- MASK_DELIMITER
- MASK_RANGE
- SUSTITUIR_CON_RANDOM_BETWEEN
- SUSTITUIR_CON_DATE_RANDOM_BETWEEN
- SHUFFLE_ROWS

CRYPTOGRAPHIC_HASH

Aplica un algoritmo a los valores de hash de la columna.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `secretId`: el ARN de la clave secreta de Secrets Manager. La clave utilizada en el algoritmo de prefijo del código de autenticación de mensajes (HMAC) basado en hash para codificar las columnas de origen, o `databrew!default` es la salida decodificada en base64 para el valor de la clave secreta de Secrets Manager.
- `secretVersion`: opcional. De forma predeterminada, es la última versión secreta.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.
- `createSecretIfMissing`: booleano opcional. Si es verdadero, intentará crear el secreto en nombre de la persona que llama.
- `algorithm`: el algoritmo utilizado para codificar sus datos. Valores de enumeración válidos: MD5, SHA1, SHA256, SHA512, HMAC_MD5, HMAC_SHA1, HMAC_SHA256, HMAC_SHA512

  Cada opción hace referencia a un algoritmo de hash diferente. Las opciones con el prefijo «HMAC» se refieren a un algoritmo de hash con clave y requieren el parámetro. `secretId` En el caso de las opciones sin el prefijo «HMAC», el parámetro no es obligatorio. `secretId`

  Si no proporciona un algoritmo de hash, el servicio tomará el valor predeterminado de «HMAC_SHA256".

```
{
   "sourceColumns": ["phonenumber"],
```

```
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "entityTypeFilter": ["USA_ALL"]
}
```

Al trabajar en la experiencia interactiva, además de la función del proyecto, el usuario de la consola debe tener permiso para acceder al `secretsmanager:GetSecretValue` secreto de Secrets Manager proporcionado.

Ejemplo de política:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

También puede optar por utilizar el secreto DataBrew-created predeterminado pasándolo `databrew!default` como SecretID y `createSecretIfMissing` el parámetro como true. Esto no se recomienda para la producción. Cualquier persona con AwsGlueDataBrewFullAccessPolicyese rol puede usar el secreto predeterminado.

DESCIFRAR

Puede utilizar la transformación DECRYPT para descifrar el interior de. DataBrew Sus datos también se pueden descifrar de forma externa o DataBrew con el AWS SDK de cifrado. Si el ARN de la clave de KMS proporcionado no coincide con el que se ha utilizado para cifrar la columna, se produce un error en la operación de descifrado. Para obtener más información sobre el SDK de AWS cifrado, consulte Qué es el SDK de AWS cifrado en la Guía para AWS Encryption SDK desarrolladores.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `kmsKeyArn`— La clave ARN de la clave del Servicio de administración de AWS claves que se utilizará para descifrar las columnas de origen. Para obtener más información sobre el ARN clave, consulte el ARN clave en la Guía para desarrolladores.AWS Key Management Service

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
}
```

Al trabajar en la experiencia interactiva, además del rol del proyecto, el usuario de la consola debe tener permiso para utilizar `kms:Decrypt` la `kms:GenerateDataKey` clave KMS proporcionada.

Ejemplo de política:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

DETERMINISTIC_DECRYPT

Descifra los datos cifrados con DETERMINISTIC_ENCRYPT.

Esta transformación no es operativa si el identificador secreto y la versión proporcionados no coinciden con los que se utilizaron para cifrar la columna.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `secretId`— El ARN de la clave secreta de Secrets Manager que se utilizará para descifrar las columnas de origen.
- `secretVersion`: opcional. De forma predeterminada, es la última versión secreta.

Ejemplo

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

Al trabajar en la experiencia interactiva, además del rol del proyecto, el usuario de la consola debe tener permiso para usar secretsmanager: GetSecretValue en el secreto de Secrets Manager proporcionado.

Ejemplo de política:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

DETERMINISTIC_ENCRYPT

Cifra la columna con una clave de 256 bits AES-GCM-SIV . Los datos cifrados con DETERMINISTIC_ENCRYPT solo se pueden descifrar dentro o con la transformación DETERMINISTIC_DECRYPT. DataBrew Esta transformación no usa el SDK de cifrado y, en su lugar, usa la biblioteca AWS KMS GitHub de LC AWS.AWS

Puede cifrar hasta 400 KB por celda. No conserva el tipo de datos al descifrarlos.

> **Note**
>
> Nota: No se recomienda usar un secreto durante más de un año.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `secretId`— ¡El ARN de la clave secreta de Secrets Manager que se utilizará para cifrar las columnas de origen o la recopilación de datos! predeterminado.
- `secretVersion`: opcional. De forma predeterminada, es la última versión secreta.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.
- `createSecretIfMissing`: booleano opcional. Si es verdadero, intentará crear el secreto en nombre de la persona que llama.

Ejemplo

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123",
   "entityTypeFilter": ["USA_ALL"]
}
```

Al trabajar en la experiencia interactiva, además de la función del proyecto, el usuario de la consola debe tener permiso para acceder al `secretsmanager:GetSecretValue` secreto de Secrets Manager proporcionado.

Ejemplo de política

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

ENCRIPTAR

Cifra los valores de las columnas de origen con el SDK de AWS cifrado. La transformación DECRYPT se puede utilizar para descifrar el interior de. DataBrew También puede descifrar los datos sin DataBrew utilizar el SDK de cifrado.AWS

La transformación ENCRYPT puede cifrar hasta 128 MiB por celda. Intentará conservar el formato al descifrarlo. Para conservar el tipo de datos, los metadatos del tipo de datos deben serializarse a menos de 1 KB. De lo contrario, debe establecer el parámetro `preserveDataType` en “false”. Los metadatos del tipo de datos se almacenarán en texto plano en el contexto de cifrado. Para obtener más información sobre el contexto de cifrado, consulte el contexto de cifrado en la AWS Key Management Service Guía para desarrolladores.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `kmsKeyArn`— La clave ARN de la clave del Servicio de administración de AWS claves que se utilizará para cifrar las columnas de origen. Para obtener más información sobre el ARN clave, consulte el ARN clave en la Guía para desarrolladores.AWS Key Management Service

- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.
- `preserveDataType`: booleano opcional. El valor predeterminado es true (verdadero). Si es “false”, el tipo de datos no se almacenará.

En el siguiente ejemplo, `entityTypeFilter` y `preserveDataType` son opcionales.

Ejemplo

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

Al trabajar en la experiencia interactiva, además de la función del proyecto, el usuario de la consola debe tener permiso para `kms:GenerateDataKey` utilizar la AWS KMS clave proporcionada.

Ejemplo de política:

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

Enmascara los caracteres que coinciden con un valor personalizado proporcionado.

Parameters

- `sourceColumns`— Una lista de nombres de columnas existentes.
- `maskSymbol`— Un símbolo que se utilizará para sustituir a los caracteres especificados.
- `regex`— Si es verdadero, se trata `customValue` como un patrón de expresiones regulares que coinciden.
- `customValue`— Todas las apariciones (o coincidencias de expresiones regulares) de `customValue` estarán enmascaradas en la cadena.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.

Example Ejemplo

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
        "Parameters": {
            "sourceColumns": ["company"],
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
}
```

MASK_DATE

Enmascara los componentes de una fecha con un símbolo de máscara especificado por el usuario.

Parameters

- `sourceColumns`— Una lista de nombres de columnas existentes.
- `maskSymbol`— Un símbolo que se utilizará para sustituir a los caracteres especificados.

- `redact`— Una matriz de enumeraciones de componentes de fechas para enmascarar. Valores de enumeración válidos: AÑO, MES, DÍA, HORA, MINUTO, SEGUNDO, MILISEGUNDO.
- `locale`— Etiqueta de idioma IETF BCP 47 opcional. El valor predeterminado es en. La configuración regional que se utilizará para formatear la fecha.

Example Ejemplo

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

MASK_DELIMITER

Enmascara los caracteres entre dos delimitadores con un símbolo de enmascaramiento especificado por el usuario.

Parameters

- `sourceColumns`— Una lista de nombres de columnas existentes.
- `maskSymbol`— Un símbolo que se utilizará para sustituir a los caracteres especificados.
- `startDelimiter`— Un carácter que indica dónde debe comenzar el enmascaramiento. Si se omite este parámetro, se aplicará la máscara empezando por el principio de la cadena.
- `endDelimiter`— Un carácter que indica dónde debe terminar el enmascaramiento. Si se omite este parámetro, se aplicará el enmascaramiento del StartDelimiter al final de la cadena.
- `preserveDelimiters`— Si es verdadero, aplica una máscara a los delimitadores.
- `alphabet`— Un conjunto de conjuntos de caracteres para conservarlos durante el enmascaramiento. Valores de enumeración válidos: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.

Example Ejemplo

```
// Mask string between '<' and '>', ignoring white spaces, symbols, and lowercase
 letters
{
    "RecipeAction": {
        "Operation": "MASK_DELIMITER",
        "Parameters": {
            "sourceColumns": ["name"],
            "maskSymbol": "#",
            "startDelimiter": "<",
            "endDelimiter": ">",
            "preserveDelimiters": false,
            "alphabet": ["WHITESPACE", "SYMBOLS"]
        }
    }
}
```

MASK_RANGE

Enmascara los caracteres situados entre dos posiciones con un símbolo de enmascaramiento especificado por el usuario.

Parameters

- `sourceColumns`— Una lista de nombres de columnas existentes.
- `maskSymbol`— Un símbolo que se utilizará para sustituir a los caracteres especificados.
- `start`— Un número que indica en qué posición de los caracteres debe comenzar el enmascaramiento (indexado a 0, ambos inclusive). Se permite la indexación negativa. Si se omite este parámetro, se aplicará la máscara desde el principio de la cadena hasta que se «detenga».
- `stop`— Un número que indica en qué posición debe terminar el enmascaramiento (indexado a 0, exclusivo). Se permite la indexación negativa. Si se omite este parámetro, se aplicará la máscara desde el «principio» hasta el final de la cadena.
- `alphabet`— Un conjunto de enumeraciones de conjuntos de caracteres para conservarlos durante el enmascaramiento. Valores de enumeración válidos: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Matriz opcional de tipos de entidades. Se puede usar para cifrar solo la PII detectada en la columna de texto libre.

Example Ejemplo

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

SUSTITUIR_CON_RANDOM_BETWEEN

Sustituye los valores por un número aleatorio.

Parameters

- `lowerBound`— El límite inferior del rango de números aleatorios.
- `sourceColumns`— Una lista de nombres de columnas existentes.
- `upperBound`— El límite superior del rango de números aleatorios.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
    }
}
```

SUSTITUIR_CON_DATE_RANDOM_BETWEEN

Sustituye los valores por una fecha aleatoria.

Parameters

- `startDate`— El inicio del intervalo de fechas a partir del cual se tomará una fecha aleatoria.
- `sourceColumns`— Una lista de los nombres de las columnas existentes.
- `endDate`— El final del intervalo de fechas a partir del cual se tomará una fecha aleatoria.

Example Ejemplo

```
{
    "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

Mezcla los valores de una columna determinada. La mezcla se puede producir con valores agrupados en una columna secundaria.

Parameters

- `sourceColumns`: matriz de columnas existentes.
- `groupByColumns`— Una matriz de columnas para agrupar las columnas de origen durante la mezcla.

Example Ejemplo

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
}
```

Pasos de la receta de detección y manipulación de valores atípicos

Siga estos pasos de la receta para trabajar con valores atípicos en sus datos y realizar transformaciones avanzadas en ellos.

Temas

FLAG_OUTLIERS

Devuelve una nueva columna que contiene un valor personalizable en cada fila que indica si el valor de la columna de origen es un valor atípico.

Parameters

- `sourceColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `targetColumn`— Especifica el nombre de una nueva columna en la que se van a insertar los resultados de la estrategia de evaluación de valores atípicos.
- `outlierStrategy`— Especifica el enfoque que se debe utilizar para detectar valores atípicos. Entre los valores válidos se incluyen:
  - `Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar.
  - `MODIFIED_Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la mediana en más del umbral de desviación absoluta de la mediana.
  - `IQR`— Identifica un valor como un valor atípico cuando supera el primer y el último cuartil de los datos de la columna. El rango intercuartil (IQR) mide dónde se encuentra el 50% medio de los puntos de datos.
- `threshold`— Especifica el valor umbral que se utilizará al detectar valores atípicos. El `sourceColumn` valor se identifica como un valor atípico si la puntuación que se calcula con `outlierStrategy` supera este número. El valor predeterminado es 3.

- `trueString`— Especifica el valor de cadena que se utilizará si se detecta un valor atípico. El valor predeterminado es «True».
- `falseString`— Especifica el valor de cadena que se utilizará si no se detecta ningún valor atípico. El valor predeterminado es «False».

En los ejemplos siguientes se muestra la sintaxis de una sola RecipeActionoperación. Una receta contiene al menos una RecipeStepoperación y un paso de receta contiene al menos una acción de receta. Una acción de receta ejecuta la transformación de datos que especifique. Un grupo de acciones de receta se ejecutan en orden secuencial para crear el conjunto de datos final.

JSON

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis JSON. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulte CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

YAML

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis YAML. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulta CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

ELIMINAR_VALORES ATÍPICOS

Elimina los puntos de datos que se clasifican como valores atípicos, en función de la configuración de los parámetros.

Parameters

- `sourceColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `outlierStrategy`— Especifica el enfoque que se debe utilizar para detectar valores atípicos. Entre los valores válidos se incluyen:
  - `Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar.
  - `MODIFIED_Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la mediana en más del umbral de desviación absoluta de la mediana.
  - `IQR`— Identifica un valor como un valor atípico cuando supera el primer y el último cuartil de los datos de la columna. El rango intercuartil (IQR) mide dónde se encuentra el 50% medio de los puntos de datos.
- `threshold`— Especifica el valor umbral que se utilizará al detectar valores atípicos. El `sourceColumn` valor se identifica como un valor atípico si la puntuación que se calcula con `outlierStrategy` supera este número. El valor predeterminado es 3.

- `removeType`— Especifica la forma de eliminar los datos. Los valores válidos son `DELETE_ROWS` y `CLEAR`.
- `trimValue`— Especifica si se van a eliminar todos o algunos de los valores atípicos. Este valor booleano tiene el valor predeterminado de. `FALSE`
  - `FALSE`— Elimina todos los valores atípicos
  - `TRUE`— Elimina los valores atípicos que se sitúan fuera del umbral percentil especificado en y. `minValue` `maxValue`
- `minValue`— Indica el valor percentil mínimo para el rango de valores atípicos. El rango válido es de 0 a 100.
- `maxValue`— Indica el valor percentil máximo para el rango de valores atípicos. El rango válido es de 0 a 100.

En los ejemplos siguientes se muestra la sintaxis de una sola RecipeActionoperación. Una receta contiene al menos una RecipeStepoperación y un paso de receta contiene al menos una acción de receta. Una acción de receta ejecuta la transformación de datos que especifique. Un grupo de acciones de receta se ejecutan en orden secuencial para crear el conjunto de datos final.

JSON

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis JSON. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulte CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

YAML

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis YAML. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulta CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

REEMPLAZAR VALORES ATÍPICOS

Actualiza los valores de los puntos de datos que se clasifican como valores atípicos, en función de la configuración de los parámetros.

Parameters

- `sourceColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `outlierStrategy`— Especifica el enfoque que se debe utilizar para detectar valores atípicos. Entre los valores válidos se incluyen:

  - `Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar.
  - `MODIFIED_Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la mediana en más del umbral de desviación absoluta de la mediana.
  - `IQR`— Identifica un valor como un valor atípico cuando supera el primer y el último cuartil de los datos de la columna. El rango intercuartil (IQR) mide dónde se encuentra el 50% medio de los puntos de datos.
- `threshold`— Especifica el valor umbral que se utilizará al detectar valores atípicos. El `sourceColumn` valor se identifica como un valor atípico si la puntuación que se calcula con `outlierStrategy` supera este número. El valor predeterminado es 3.
- `replaceType`— Especifica el método que se debe utilizar al reemplazar los valores atípicos. Entre los valores válidos se incluyen:
  - `WINSORIZE_VALUES`— Especifica el uso de los percentiles mínimo y máximo para limitar los 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 el tipo de función modal que se va a utilizar cuando `replaceType` es. `REPLACE_WITH_MODE` Los valores válidos incluyen los siguientes: `MINMAX`, y`AVERAGE`.
- `minValue`— Indica el valor percentil mínimo para el rango de valores atípicos que se va a aplicar cuando `trimValue` se utilice. El rango válido es de 0 a 100.
- `maxValue`— Indica el valor percentil máximo para el rango de valores atípicos que se aplicará cuando se utilice. `trimValue` El rango válido es de 0 a 100.
- `value`— Especifica el valor que se debe insertar cuando se utilice. `REPLACE_WITH_CUSTOM`
- `trimValue`— Especifica si se van a eliminar todos o algunos de los valores atípicos. Este valor booleano se establece en `TRUE` when `replaceType` is`REPLACE_WITH_NULL`,

`REPLACE_WITH_MODE` o. `WINSORIZE_VALUES` El valor predeterminado es para `FALSE` todos los demás.

- `FALSE`— Elimina todos los valores atípicos
- `TRUE`—Elimina los valores atípicos que se sitúan fuera del límite máximo de percentiles especificado en y. `minValue` `maxValue`

En los ejemplos siguientes se muestra la sintaxis de una sola operación. RecipeAction Una receta contiene al menos una RecipeStepoperación y un paso de receta contiene al menos una acción de receta. Una acción de receta ejecuta la transformación de datos que especifique. Un grupo de acciones de receta se ejecutan en orden secuencial para crear el conjunto de datos final.

JSON

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis JSON. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulte CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

YAML

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis YAML. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulta CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

REESCALE_OUTLIERS_WITH_Z_SCORE

Devuelve una nueva columna con un valor atípico reescalado en cada fila, en función de la configuración de los parámetros. Esta acción también aplica Z-score la normalización a los valores de datos escalados linealmente para que tengan una media (μ) de 0 y una desviación estándar (σ) de 1. Recomendamos esta acción para gestionar valores atípicos.

Parameters

- `sourceColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `targetColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `outlierStrategy`— Especifica el enfoque que se debe utilizar para detectar valores atípicos. Entre los valores válidos se incluyen:

  - `Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar.
  - `MODIFIED_Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la mediana en más del umbral de desviación absoluta de la mediana.
  - `IQR`— Identifica un valor como un valor atípico cuando supera el primer y el último cuartil de los datos de la columna. El rango intercuartil (IQR) mide dónde se encuentra el 50% medio de los puntos de datos.
- `threshold`— El valor umbral que se utilizará al detectar valores atípicos. El `sourceColumn` valor se identifica como un valor atípico si la puntuación que se calcula con `outlierStrategy` supera este número. El valor predeterminado es 3.

En los ejemplos siguientes se muestra la sintaxis de una sola RecipeActionoperación. Una receta contiene al menos una RecipeStepoperación y un paso de receta contiene al menos una acción de receta. Una acción de receta ejecuta la transformación de datos que especifique. Un grupo de acciones de receta se ejecutan en orden secuencial para crear el conjunto de datos final.

JSON

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de una operación de DataBrew receta, con la sintaxis JSON. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulte CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

YAML

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de una operación de DataBrew receta, con la sintaxis YAML. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en YAML

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: Z_SCORE
    threshold: '3'
```

Para obtener más información sobre el uso de esta acción de receta en una operación de API, consulta CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

CAMBIAR LA ESCALA DE VALORES ATÍPICOS CON UN SESGO

Devuelve una nueva columna con un valor atípico reescalado en cada fila, en función de la configuración de los parámetros. Esta acción sirve para reducir la asimetría de la distribución mediante la aplicación de la transformación logarítmica o raíz especificada. Se recomienda esta acción para gestionar datos asimétricos.

Parameters

- `sourceColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `targetColumn`— Especifica el nombre de una columna numérica existente que puede contener valores atípicos.
- `outlierStrategy`— Especifica el enfoque que se debe utilizar para detectar valores atípicos. Entre los valores válidos se incluyen:

  - `Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la media por encima del umbral de desviación estándar.
  - `MODIFIED_Z_SCORE`— Identifica un valor como un valor atípico cuando se desvía de la mediana en más del umbral de desviación absoluta de la mediana.
  - `IQR`— Identifica un valor como un valor atípico cuando supera el primer y el último cuartil de los datos de la columna. El rango intercuartil (IQR) mide dónde se encuentra el 50% medio de los puntos de datos.
- `threshold`— Especifica el valor umbral que se utilizará al detectar valores atípicos. El `sourceColumn` valor se identifica como un valor atípico si la puntuación que se calcula con `outlierStrategy` supera este número. El valor predeterminado es 3.
- `skewFunction`— Especifica el método que se debe utilizar al reemplazar los valores atípicos. Entre los valores válidos se incluyen:
  - LOG: aplica una transformación fuerte para reducir el sesgo positivo y negativo. Se trata de un logaritmo natural (2,718281828).
  - RAÍZ (`convalue = 3`): aplica una transformación bastante fuerte para reducir el sesgo positivo y negativo. (Raíz cúbica)
  - RAÍZ (`convalue = 2`): aplica una transformación moderada para reducir únicamente el sesgo positivo. (Raíz cuadrada)
  - CUADRADO: aplica una transformación moderada para reducir el sesgo negativo. (Cuadrado)
  - Transformación personalizada: aplica la `ROOT` transformación `LOG` o la especificada mediante el número personalizado proporcionado en el `value` parámetro.
- `value`— Especifica el valor que se va a utilizar para la transformación personalizada. Si `skewFunction` es LOG, este valor representa la base del registro. Si `skewFunction` es ROOT, este valor representa la potencia de la raíz.

En los ejemplos siguientes se muestra la sintaxis de una sola RecipeActionoperación. Una receta contiene al menos una RecipeStepoperación y un paso de receta contiene al menos una acción de receta. Una acción de receta ejecuta la transformación de datos que especifique. Un grupo de acciones de receta se ejecutan en orden secuencial para crear el conjunto de datos final.

JSON

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis JSON. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulte CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

YAML

A continuación, se muestra un ejemplo `RecipeAction` para usarlo como miembro de un `RecipeStep` ejemplo de DataBrew receta, con la sintaxis YAML. Para ver ejemplos de sintaxis que muestran una lista de acciones de recetas, consulteDefinir la estructura de una receta.

Example Ejemplo en 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 obtener más información sobre el uso de esta acción de receta en una operación de API, consulta CreateRecipeo UpdateRecipe. Puedes usar estas y otras operaciones de la API en tu propio código.

Pasos de la receta de estructura de columnas

Utilice estos pasos de la receta de estructura de columnas para modificar la estructura de columnas de sus datos.

Temas

OPERACIÓN_BOOLEANA

Cree una nueva columna en función del resultado de la condición lógica IF. Devuelve el valor verdadero si la expresión booleana es verdadera, el valor falso si la expresión booleana es falsa o devuelve un valor personalizado.

Parameters

- `trueValueExpression`— Resultado cuando se cumple la condición.
- `falseValueExpression`— Resultado cuando no se cumple la condición.
- `valueExpression`— Condición booleana.
- `withExpressions`— Configuración para resultados agregados.
- `targetColumn`: un nombre para la columna recién creada.

Puede utilizar valores constantes, referencias a columnas y resultados agregados en trueValueExpression, false ValueExpression y ValueExpression.

Example Ejemplo: valores constantes

Valores que permanecen sin cambios, como un número o una oración.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Ejemplo: referencias a columnas

Valores que son columnas del conjunto de datos.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Ejemplo: resultados agregados

Valores que se calculan mediante funciones de agregación. Una función de agregado realiza un cálculo en una columna y devuelve un 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"
      }
    }
  }
}
```

Los usuarios deben convertir el JSON en una cadena escapando.

Tenga en cuenta que los nombres de los parámetros en true ValueExpressionValueExpression, false y ValueExpression deben coincidir con los nombres de WithExpressions. Para utilizar los resultados agregados de algunas columnas, debe crear parámetros para ellas y proporcionar las funciones de agregado.

Example Ejemplo:

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
```

```
      }
    }
  }
}
```

Example Ejemplo: and/or

Puede usar y o para combinar varias condiciones.

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

Funciones de agregación válidas

La siguiente tabla muestra todas las funciones de agregado válidas que se pueden usar en una operación booleana.

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
Numérico	Sum	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Devuelve la suma de `column.1`
	Media	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" } ]`	Devuelve la media de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Desviación absoluta media	`:desviación absoluta media. Column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Devuelve la desviación absoluta media de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Median	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Devuelve la mediana de `column.1`
	Producto	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Devuelve el producto de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Desviación estándar	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Devuelve la desviación estándar de `column.1`
	Varianza	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Devuelve la varianza de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Error estándar de la media	`:error estándar de mean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Devuelve el error estándar de la media de `column.1`
	Asimetría	`:asimetría.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Devuelve la asimetría de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Curtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Devuelve la curtosis de `column.1`
Datetime/ Numeric/Text	Recuento	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Devuelve el número total de filas de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Recuento distinto	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Devuelve el número total de filas distintas de `column.1`
	Mínimo	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Devuelve el valor mínimo de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Máximo	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Devuelve el valor máximo de `column.1`

Condiciones válidas en una ValueExpression

En la siguiente tabla se muestran las condiciones admitidas y las expresiones de valores que puede utilizar.

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
Cadena	Contiene	contiene (`columna`, 'texto')	Condición para comprobar si el valor de la columna contiene texto
	No contiene	! contiene (`columna`, 'texto')	Condición para comprobar si el valor de la columna no contiene texto

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Coincide	coincidencias (`columna`, 'patrón')	Condición para comprobar si el valor de la columna coincide con el patrón
	No coincide	! coincidencias (`columna`, 'patrón')	Condición para comprobar si el valor de la columna no coincide con el patrón
	Empieza por	StartsWith (`columna`, 'texto')	Condición para comprobar si el valor de la columna comienza con texto
	No comienza con	! StartsWith (`columna`, 'texto')	Condición para comprobar si el valor de la columna no comienza con texto
	Acaba con	Termina con (`columna`, 'texto')	Condición para comprobar si el valor de la columna termina con texto
	No termina en	! Termina con (`columna`, 'texto')	Condición para comprobar si el valor de la columna no termina con texto
Numérico	Menor que	`columna` < número	Condición para comprobar si el valor de la columna es menor que un número

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Menor o igual que	`columna` <= número	Condición para comprobar si el valor de la columna es menor o igual que un número
	Mayor que	`columna` > número	Condición para comprobar si el valor de la columna es mayor que un número
	Mayor o igual que	`columna` >= número	Condición para comprobar si el valor de la columna es mayor o igual que un número
	Está entre	isBetween (`column`, minNumber, maxNumber)	Condición para comprobar si el valor de la columna está entre minNumber y maxNumber
	No está entre	! isBetween (`columna`, número mínimo, número máximo)	Condición para comprobar si el valor de la columna no está entre minNumber y maxNumber
Booleano	¿Es cierto	`column` = VERDADERO	Condición para comprobar si el valor de la columna es booleano TRUE

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Es falso	`column` = FALSO	Condición para comprobar si el valor de la columna es booleano FALSO
Date/Timestamp	Antes de	`column` < 'fecha'	Condición para comprobar si el valor de la columna es anterior a la fecha
	Anterior o igual a	`columna` <= 'fecha'	Condición para comprobar si el valor de la columna es anterior o igual a la fecha
	Más tarde que	`columna` > 'fecha'	Condición para comprobar si el valor de la columna es posterior a la fecha
	Mayor o igual a	`columna` >= 'fecha'	Condición para comprobar si el valor de la columna es posterior o igual a la fecha
String/Numeric/Date/Timestamp	Es exactamente	`column` = 'valor'	Condición para comprobar si el valor de la columna es exactamente el valor

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Is not	`columna`! = 'valor'	Condición para comprobar si el valor de la columna no es un valor
	Falta	Falta (`column`)	Condición para comprobar si falta el valor de la columna
	No falta	! Falta (`column`)	Condición para comprobar si no falta el valor de la columna
	¿Es válido	IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
	No es válido	! IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna no es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
Anidado	Falta	Falta (`column`)	Condición para comprobar si falta el valor de la columna

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	No falta	! Falta (`column`)	Condición para comprobar si no falta el valor de la columna
	¿Es válido	IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
	No es válido	! IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna no es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)

CASE_OPERATION

Cree una nueva columna en función del resultado de la condición lógica CASE. La operación de caso examina las condiciones de cada caso y devuelve un valor cuando se cumple la primera condición. Una vez que se cumple una condición, la operación deja de leer y devuelve el resultado. Si no se cumple ninguna condición, devuelve el valor predeterminado.

Parameters

- `valueExpression`— Condiciones.
- `withExpressions`— Configuración para resultados agregados.
- `targetColumn`— Nombre de la columna recién creada.

Example Ejemplo

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

Funciones de agregación válidas

La siguiente tabla muestra todas las funciones de agregación válidas que se pueden utilizar en una operación de caso.

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
Numérico	Sum	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Devuelve la suma de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Media	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" } ]`	Devuelve la media de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Desviación absoluta media	`:desviación absoluta media. Column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Devuelve la desviación absoluta media de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Median	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Devuelve la mediana de `column.1`
	Producto	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Devuelve el producto de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Desviación estándar	`:standarddeviation.column.1`	`[ { "name": "standard deviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Devuelve la desviación estándar de `column.1`
	Varianza	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Devuelve la varianza de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Error estándar de la media	`:error estándar de mean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Devuelve el error estándar de la media de `column.1`
	Asimetría	`:asimetría.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Devuelve la asimetría de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Curtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Devuelve la curtosis de `column.1`
Datetime/ Numeric/Text	Recuento	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Devuelve el número total de filas de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Recuento distinto	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Devuelve el número total de filas distintas de `column.1`
	Mínimo	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Devuelve el valor mínimo de `column.1`

Tipo de columna	Condición	Expresión de valor	Con expresiones	Valor devuelto
	Máximo	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Devuelve el valor máximo de `column.1`

Condiciones válidas en una ValueExpression

En la siguiente tabla se muestran las condiciones admitidas y las expresiones de valores que puede utilizar.

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
Cadena	Contiene	contiene (`columna`, 'texto')	Condición para comprobar si el valor de la columna contiene texto
	No contiene	! contiene (`columna`, 'texto')	Condición para comprobar si el valor de la columna no contiene texto

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Coincide	coincidencias (`columna`, 'patrón')	Condición para comprobar si el valor de la columna coincide con el patrón
	No coincide	! coincidencias (`columna`, 'patrón')	Condición para comprobar si el valor de la columna no coincide con el patrón
	Empieza por	StartsWith (`columna`, 'texto')	Condición para comprobar si el valor de la columna comienza con texto
	No comienza con	! StartsWith (`columna`, 'texto')	Condición para comprobar si el valor de la columna no comienza con texto
	Acaba con	Termina con (`columna`, 'texto')	Condición para comprobar si el valor de la columna termina con texto
	No termina en	! Termina con (`columna`, 'texto')	Condición para comprobar si el valor de la columna no termina con texto
Numérico	Menor que	`columna` < número	Condición para comprobar si el valor de la columna es menor que un número

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Menor o igual que	`columna` <= número	Condición para comprobar si el valor de la columna es menor o igual que un número
	Mayor que	`columna` > número	Condición para comprobar si el valor de la columna es mayor que un número
	Mayor o igual que	`columna` >= número	Condición para comprobar si el valor de la columna es mayor o igual que un número
	Está entre	isBetween (`column` , minNumber, maxNumber)	Condición para comprobar si el valor de la columna está entre minNumber y maxNumber
	No está entre	! isBetween (`columna `, número mínimo, número máximo)	Condición para comprobar si el valor de la columna no está entre minNumber y maxNumber
Booleano	¿Es cierto	`column` = VERDADERO	Condición para comprobar si el valor de la columna es booleano TRUE

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Es falso	`column` = FALSO	Condición para comprobar si el valor de la columna es booleano FALSO
Date/Timestamp	Antes de	`column` < 'fecha'	Condición para comprobar si el valor de la columna es anterior a la fecha
	Anterior o igual a	`columna` <= 'fecha'	Condición para comprobar si el valor de la columna es anterior o igual a la fecha
	Más tarde que	`columna` > 'fecha'	Condición para comprobar si el valor de la columna es posterior a la fecha
	Mayor o igual a	`columna` >= 'fecha'	Condición para comprobar si el valor de la columna es posterior o igual a la fecha
String/Numeric/Date/Timestamp	Es exactamente	`column` = 'valor'	Condición para comprobar si el valor de la columna es exactamente el valor

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	Is not	`columna`! = 'valor'	Condición para comprobar si el valor de la columna no es un valor
	Falta	Falta (`column`)	Condición para comprobar si falta el valor de la columna
	No falta	! Falta (`column`)	Condición para comprobar si no falta el valor de la columna
	¿Es válido	IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
	No es válido	! IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna no es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
Anidado	Falta	Falta (`column`)	Condición para comprobar si falta el valor de la columna

Tipo de columna	Condición	Expresión de valor	Description (Descripción)
	No falta	! Falta (`column`)	Condición para comprobar si no falta el valor de la columna
	¿Es válido	IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)
	No es válido	! IsValid (`column`, tipo de datos)	Condición para comprobar si el valor de la columna no es válido (el valor es de tipo de datos o se puede convertir a tipo de datos)

FLAG_COLUMN_FROM_NULL

Crea una nueva columna en función de la presencia de valores nulos en una columna existente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`— El nombre de la nueva columna que se va a crear.
- `flagType`— Un valor que se debe establecer en`Null values`.
- `trueString`— Un valor para la nueva columna, si se encuentra un valor nulo en la fuente. Si no se especifica ningún valor, el valor predeterminado es `True`.
- `falseString`— Un valor para la nueva columna, si se encuentra un valor no nulo en la fuente. Si no se especifica ningún valor, el valor predeterminado es `False`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

Crea una nueva columna en función de la presencia de un patrón especificado por el usuario en una columna existente.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`— El nombre de la nueva columna que se va a crear.
- `flagType`— Un valor que se debe establecer en`Pattern`.
- `pattern`— Una expresión regular que indica el patrón que se va a evaluar.
- `trueString`— Un valor para la nueva columna, si se encuentra un valor nulo en la fuente. Si no se especifica ningún valor, el valor predeterminado es `True`.
- `falseString`— Un valor para la nueva columna, si se encuentra un valor no nulo en la fuente. Si no se especifica ningún valor, el valor predeterminado es `False`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
```

```
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
    }
}
```

MERGE

Combina dos o más columnas en una nueva columna.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de una o más columnas que se van a fusionar.
- `delimiter`— Un separador opcional entre los valores, que aparecerá en la columna de destino.
- `targetColumn`— El nombre de la columna combinada que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_DELIMITER

Divide una columna en tres columnas nuevas, según un delimitador inicial y uno final.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `patternOption1`— Una JSON-encoded cadena que representa uno o más caracteres que indican el primer delimitador.
- `patternOption2`— Una JSON-encoded cadena que representa uno o más caracteres que indican el segundo delimitador.
- `pattern`— Uno o más caracteres para usarlos como separadores al dividir los datos.
- `includeInSplit`— Si es verdadero, incluye el patrón en la nueva columna; de lo contrario, el patrón se descarta.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_DELIMITER",
        "Parameters": {
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

SPLIT_COLUMN_ENTRE_POSICIONES

Divide una columna en tres columnas nuevas, según los desfases que especifique.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `startPosition`— La posición del personaje en la que va a empezar la división.
- `endPosition`— La posición del personaje en la que va a terminar la división.

Example Ejemplo

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

```
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
        }
    }
}
```

SPLIT_COLUMN_FROM_END

Divide una columna en dos columnas nuevas, con un desplazamiento respecto al final de la cadena.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `position`— La posición del carácter, desde el extremo derecho de la cadena, en la que se producirá la división.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
}
```

SPLIT_COLUMN_FROM_START

Divide una columna en dos columnas nuevas, con un desplazamiento respecto al principio de la cadena.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `position`— La posición del carácter, desde el extremo izquierdo de la cadena, en la que se producirá la división.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

SPLIT_COLUMN_MULTIPLE_DELIMITER

Divide una columna según varios delimitadores.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `patternOptions`— Una JSON-encoded cadena que representa uno o más patrones que determinan los criterios de división.
- `pattern`— Uno o más caracteres para usarlos como separadores al dividir los datos.
- `limit`— Cuántas divisiones realizar. El mínimo es 1; el máximo es 20.
- `includeInSplit`— Si es verdadero, incluye el patrón en la nueva columna; de lo contrario, el patrón se descarta.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
```

```
}
```

SPLIT_COLUMN_SINGLE_DELIMITER

Divide una columna en una o más columnas nuevas, según un delimitador específico.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— Uno o más caracteres para usarlos como separadores al dividir los datos.
- `limit`— Cuántas divisiones realizar. El mínimo es 1; el máximo es 20.
- `includeInSplit`— Si es verdadero, incluye el patrón en la nueva columna; de lo contrario, el patrón se descarta.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_SINGLE_DELIMITER",
        "Parameters": {
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVALS

Divide una columna a intervalos de n caracteres, especificando n.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `startPosition`— La posición de los caracteres en la que va a comenzar la división.
- `interval`— El número de caracteres que se van a omitir antes de la siguiente división.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

Pasos de la receta de formato de columnas

Siga los pasos de la receta de formato de columnas para cambiar el formato de los datos de las columnas.

Temas

- FORMATO_NUMÉRICO
- FORMAT_NÚMERO_DE_TELÉFONO

FORMATO_NUMÉRICO

Devuelve una columna en la que un valor numérico se convierte en una cadena formateada.

Parameters

- `sourceColumn` – Cadena. El nombre de una columna existente.
- `decimalPlaces`— Entero. El valor del número de dígitos después del separador decimal.
- `numericDecimalSeparator` – Cadena. Uno de los siguientes valores que indica el separador decimal:
  - "."
  - ","
- `numericThousandSeparator` – Cadena. Uno de los siguientes valores que indica el separador de miles:

  - nulo. Indica que el separador de miles no está activado.
  - ","
  - " "
  - "."
  - "\\"
- `numericAbbreviatedUnit` – Cadena. Uno de los siguientes valores que indica la unidad de abreviatura:
  - nulo. Indica que una unidad de abreviatura no está habilitada.
  - «MIL»
  - «MILLÓN»
  - «MIL MILLONES»
  - «BILLÓN»
- `numericUnitAbbreviation` – Cadena. Uno de los valores siguientes o cualquier valor personalizado, que indique la abreviatura de la unidad:
  - nulo. Indica que la abreviatura de unidades no está habilitada.
  -

Unidad de abreviatura	Opciones
Miles	K, k, M, mil, personalizado
Million	M, m, MM, millón, personalizado
Billion	B, bn, mil millones, personalizado
Trillón	T, diez, trillones, personalizado

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
```

```
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
            "numericUnitAbbreviation": "K"
        }
    }
}
```

FORMAT_NÚMERO_DE_TELÉFONO

Devuelve una columna en la que la cadena de un número de teléfono se convierte en un valor formateado.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `phoneNumberFormat`: el formato al que se va a convertir el número de teléfono. Si no se especifica ningún formato, el formato predeterminado es `E.164`, un formato de número de teléfono estándar reconocido internacionalmente. Entre los valores válidos se incluyen:
  - `E164`(omita el punto siguiente) `E`
- `defaultRegion`: un código de región válido compuesto por dos o tres letras mayúsculas que especifica la región del número de teléfono cuando no hay ningún código de país en el propio número. Como máximo, se puede proporcionar uno de `defaultRegion` o `defaultRegionColumn`.
- `defaultRegionColumn`— El nombre de una columna del tipo `Country` de datos avanzado. El código de región de la columna especificada se utiliza para determinar el código de país del número de teléfono cuando no hay ningún código de país en el propio número. Como máximo, se puede proporcionar uno de `defaultRegion` o `defaultRegionColumn`.

Notas

- Las entradas a las que no se pueda dar formato a un número de teléfono válido permanecen sin modificar.
- Si no se proporciona una región predeterminada y un número de teléfono no comienza con un símbolo más (+) ni un prefijo de país, el número de teléfono no está formateado.

Example

Ejemplo: región predeterminada fija

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

Ejemplo: opción de columna de región predeterminada

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

Pasos de la receta de estructura de datos

Utilice estos pasos de la receta para tabular y resumir los datos desde diferentes perspectivas o para realizar funciones avanzadas.

Temas

- NEST_TO_ARRAY
- NEST_TO_MAP
- NEST_TO_STRUCT
- UNNEST_ARRAY
- UNNEST_MAP
- UNNEST_STRUCT

- UNNEST_STRUCT_N
- GROUP_BY
- JOIN
- PIVOT
- SCALE
- TRANSPONER
- UNION
- UNPIVOT

NEST_TO_ARRAY

Convierte las columnas seleccionadas por el usuario en valores de matriz. El orden de las columnas seleccionadas se mantiene al crear la matriz resultante. Los distintos tipos de datos de las columnas se encasillan en un tipo común que admite los tipos de datos de todas las columnas.

Parameters

- `sourceColumns`— Lista de las columnas de origen.
- `targetColumn`— El nombre de la columna de destino.
- `removeSourceColumns`— Contiene el valor `true` o `false` para indicar si el usuario desea o no eliminar las columnas de origen seleccionadas.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_MAP

Convierte las columnas seleccionadas por el usuario en pares clave-valor, cada uno con una clave que representa el nombre de la columna y un valor que representa el valor de la fila. El orden de la columna seleccionada no se mantiene al crear el mapa resultante. Los distintos tipos de datos de las columnas están encasillados en un tipo común que admite los tipos de datos de todas las columnas.

Parameters

- `sourceColumns`— Lista de las columnas de origen.
- `targetColumn`— El nombre de la columna de destino.
- `removeSourceColumns`— Contiene el valor `true` o `false` para indicar si el usuario desea o no eliminar las columnas de origen seleccionadas.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_STRUCT

Convierte las columnas seleccionadas por el usuario en pares clave-valor, cada uno con una clave que representa el nombre de la columna y un valor que representa el valor de la fila. El orden de las columnas seleccionadas y el tipo de datos de cada columna se mantienen en la estructura resultante.

Parameters

- `sourceColumns`— Lista de las columnas de origen.
- `targetColumn`— El nombre de la columna de destino.

- `removeSourceColumns`— Contiene el valor `true` o `false` para indicar si el usuario desea o no eliminar las columnas de origen seleccionadas.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

UNNEST_ARRAY

Desanida una columna de tipo `array` en una nueva columna. Si la matriz contiene más de un valor, se genera una fila correspondiente a cada elemento. Esta función solo deshace un nivel de una columna de matriz.

Parameters

- `sourceColumn`— El nombre de una columna existente. Esta columna debe ser de este `struct` tipo.
- `targetColumn`— Nombre de la columna de destino que se genera.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
```

```
}
```

UNNEST_MAP

Desanida una columna de tipo `map` y genera una columna para la clave y el valor. Si hay más de un par clave-valor, se generará una fila correspondiente a cada valor clave. Esta función solo deshace un nivel de una columna del mapa.

Parameters

- `sourceColumn`— El nombre de una columna existente. Esta columna debe ser de este `struct` tipo.
- `removeSourceColumn`— Si`true`, la columna de origen se elimina una vez finalizada la función.
- `targetColumn`— Si se proporciona, cada una de las columnas generadas empezará con este prefijo.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

Desanida una columna de tipo `struct` y genera una columna para cada una de las claves presentes en la estructura. Esta función solo deshace la estructura de nivel uno.

Parameters

- `sourceColumn`— El nombre de una columna existente. Esta columna debe ser de tipo estructural.
- `removeSourceColumn`— Si`true`, la columna de origen se elimina una vez finalizada la función.

- `targetColumn`— Si se proporciona, cada una de las columnas generadas empezará con este prefijo.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

Crea una nueva columna para cada campo del tipo de columna seleccionado. `struct`

Por ejemplo, dada la siguiente estructura:

```
           user {
              name: “Ammy”
              address: {
                 state: "CA",
                 zipcode: 12345
              }
           }
```

Esta función crea 3 columnas:

nombre de usuario	user.address.state	user.address.zip code
Ammy	CA	12345

Parameters

- `sourceColumns`— Lista de las columnas de origen.
- `regexColumnSelector`— Una expresión regular para seleccionar las columnas que se van a anidar.
- `removeSourceColumn`— Un valor booleano. Si es verdadero, elimina la columna de origen; de lo contrario, consérvala.
- `unnestLevel`— El número de niveles que se van a deshacer.
- `delimiter`— El delimitador se utiliza en el nombre de la columna recién creada para separar los diferentes niveles de la estructura. Por ejemplo: si el delimitador es «/», el nombre de la columna tendrá este formato: «user/address/state».
- `conditionExpressions`— Expresiones de condición.

Example Ejemplo

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

GROUP_BY

Resume los datos agrupando las filas en una o más columnas y, a continuación, aplicando una función de agregación a cada grupo.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas que forman la base de cada grupo.
- `groupByAggFunctions`— Una JSON-encoded cadena que representa una lista de funciones de agregación que se van a aplicar. (Si no desea la agregación, especifique)`UNAGGREGATED`.

- `useNewDataFrame`— Si es verdadero, los resultados de GROUP_BY están disponibles en la sesión del proyecto y sustituyen a su contenido actual.

Example Ejemplo

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
      }
    }
  }
]
```

JOIN

Realiza una operación de unión en dos conjuntos de datos.

Parameters

- `joinKeys`— Una JSON-encoded cadena que representa una lista de columnas de cada conjunto de datos para que actúen como claves de unión.
- `joinType`— El tipo de unión que se va a realizar. Debe ser uno de los siguientes: `INNER_JOIN` `LEFT_JOIN` | `RIGHT_JOIN` | `OUTER_JOIN` | `LEFT_EXCLUDING_JOIN` | `RIGHT_EXCLUDING_JOIN` | `OUTER_EXCLUDING_JOIN`
- `leftColumns`— Una JSON-encoded cadena que representa una lista de columnas del conjunto de datos activo actual.
- `rightColumns`— Una JSON-encoded cadena que representa una lista de columnas de otro conjunto de datos (secundario) para unirlas al conjunto actual.
- `secondInputLocation`— Una URL de Amazon S3 que se convierte en el archivo de datos del conjunto de datos secundario.
- `secondaryDatasetName`— El nombre del conjunto de datos secundario.

Example Ejemplo

```
{
    "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

Convierte todos los valores de fila de una columna seleccionada en columnas individuales con valores.

Parameters

- `sourceColumn`— El nombre de una columna existente. La columna puede tener un máximo de 10 valores distintos.
- `valueColumn`— El nombre de una columna existente. La columna puede tener un máximo de 10 valores distintos.
- `aggregateFunction`— El nombre de una función de agregación. Si no desea la agregación, utilice la palabra clave`COLLECT_LIST`.

Example Ejemplo

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
            "valueColumn": "all_votes"
        }
    }
}
```

SCALE

Escala o normaliza el rango de datos de una columna numérica.

Parameters

- `sourceColumn`— El nombre de una columna existente.
- `strategy`— La operación que se aplicará a los valores de las columnas:
  - `MIN_MAX`— Cambia la escala de los valores a un rango de [0,1].
  - `SCALE_BETWEEN`— Cambia la escala de los valores a un rango de dos valores específicos.
  - `MEAN_NORMALIZATION`— Cambia la escala de los datos para que tengan una media (μ) de 0 y una desviación estándar (σ) de 1 dentro de un rango de [-1, 1].
  - `Z_SCORE`— Escala linealmente los valores de los datos para que tengan una media (μ) de 0 y una desviación estándar (σ) de 1. Ideal para tratar valores atípicos.
- `targetColumn`— El nombre de la columna que contiene los resultados.

Example Ejemplo

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
```

```
        }
    }
}
```

TRANSPONER

Convierte todas las filas seleccionadas en columnas y las columnas en filas.

Column 1	Column A	Column B	Column C
Row A	Value A	Value B	Value C
Row B	Value A1	Value B1	Value C1

New column	Row A	Row B
Column A	Value A	Value A1
Column B	Value B	Value B1
Column C	Value C	Value C1

Parameters

- `pivotColumns`— Una JSON-encoded cadena que representa una lista de columnas cuyas filas se convertirán en nombres de columnas.
- `valueColumns`— Una JSON-encoded cadena que representa una lista de una o más columnas que se van a convertir en filas.
- `aggregateFunction`— El nombre de una función de agregación. Si no desea la agregación, utilice la palabra clave`COLLECT_LIST`.
- `newColumn`— La columna que contiene las columnas transpuestas como valores.

Example Ejemplo

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
```

```
            "newColumn": "Student"
        }
    }

}
```

UNION

Combina las filas de dos o más conjuntos de datos en un único resultado.

Parameters

- `datasetsColumns`— Una JSON-encoded cadena que representa una lista de todas las columnas de los conjuntos de datos.
- `secondaryDatasetNames`— Una JSON-encoded cadena que representa una lista de uno o más conjuntos de datos secundarios.
- `secondaryInputs`— Una JSON-encoded cadena que representa una lista de buckets de Amazon S3 y nombres de claves de objetos que indican DataBrew dónde encontrar los conjuntos de datos secundarios.
- `targetColumnNames`— Una JSON-encoded cadena que representa una lista de nombres de columnas para los resultados.

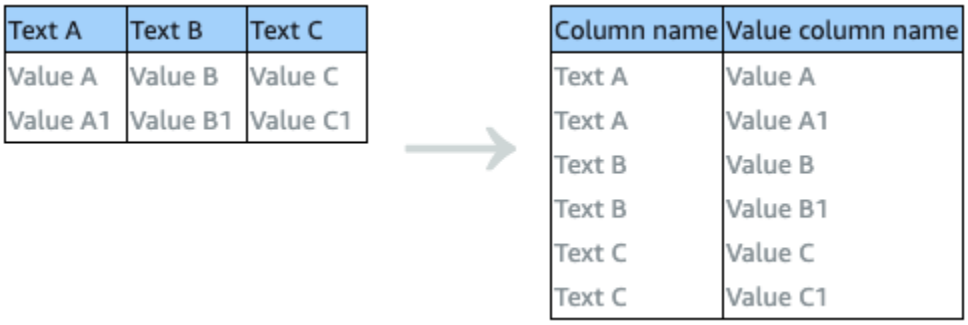
Example Ejemplo

```
{
    "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]]",
            "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

Convierte todos los valores de las columnas de una fila seleccionada en filas individuales con 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

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de una o más columnas que no se pueden pivotar.
- `unpivotColumn`— La columna de valores de la operación de despivote.
- `valueColumn`— La columna que contiene los valores no pivotantes.

Example Ejemplo

```
{
    "Action": {
        "Operation": "UNPIVOT",
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
    }
}
```

Pasos de la receta de ciencia de datos

Utilice estos pasos de la receta para tabular y resumir los datos desde diferentes perspectivas o para realizar transformaciones avanzadas.

Temas

BINARIZACIÓN

Toma todos los valores de una columna de origen numérico seleccionada, los compara con un valor umbral y genera una nueva columna con un 1 o un 0 para cada fila.

Parameters

- `sourceColumn`: el nombre de una columna existente.

  `targetColumn`: el nombre de la nueva columna que se va a crear.

  `threshold`— Número que indica el umbral para asignar el valor de 0 o 1.

  `flip`— Opción para invertir la asignación binaria de modo que a los valores más bajos se les asigne 1 y a los valores más altos se les asigne 0. Cuando el parámetro flip es verdadero, los valores inferiores o iguales al valor umbral dan como resultado 1 y los valores superiores al valor umbral dan como resultado 0.

Example Ejemplo

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

```
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
}
```

AGRUPAMIENTO

La agrupación en grupos (denominada agrupamiento en bloques en la consola) toma los elementos de una columna de valores numéricos, los agrupa en grupos definidos por rangos numéricos y genera una nueva columna que muestra el intervalo de cada fila. La división en cubos se puede realizar mediante divisiones o porcentajes. En el primer ejemplo que aparece a continuación se utilizan divisiones y en el segundo se utiliza un porcentaje.

Parameters

- `sourceColumn`: el nombre de una columna existente.

  `targetColumn`: el nombre de la nueva columna que se va a crear.

  `bucketNames`— Lista de nombres de cubos.

  `splits`— Lista de niveles de cubos. Los cubos son consecutivos y el límite superior de un cubo será el límite inferior del siguiente cubo.

  `percentage`— Cada cubo se describirá como un porcentaje.

Example Ejemplo de uso de divisiones

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
```

```
        }
    }
}
```

Example Ejemplo de uso de un porcentaje

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

MAPEO CATEGÓRICO

Asigna uno o más valores categóricos a valores numéricos o de otro tipo

Parameters

- `sourceColumn`: el nombre de una columna existente.

  `categoryMap`— Una JSON-encoded cadena que representa un mapa de valores a categorías.

  `deleteOtherRows`— Si`true`, todas las filas no mapeadas se eliminarán del conjunto de datos.

  `other`— Cuando se proporcione, todos los valores no mapeados se sustituirán por este valor.

  `keepOthers`— Si es verdadero, todos los valores no mapeados seguirán siendo los mismos.

  `mapType`— El tipo de datos de la columna mapeada.

  `targetColumn`— El nombre de la columna que contiene los resultados.

Example Ejemplo

```
{
```

```
    "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

Crea n columnas numéricas, donde n es el número de valores únicos de una variable categórica seleccionada.

Por ejemplo, considere una columna llamada`shirt_size`. Las camisas están disponibles en talla pequeña, mediana, grande o extra grande. Los datos de la columna pueden tener el siguiente aspecto.

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
M
```

En este escenario, hay cuatro valores distintos para`shirt_size`. Por lo tanto, `ONE_HOT_ENCODING` genera cuatro columnas nuevas. Cada nueva columna recibe un nombre`shirt_size_`*x*, donde *x* representa un `shirt_size` valor distinto.

Los resultados de las cuatro columnas generadas `shirt_size` y las cuatro columnas generadas tienen este aspecto.

```
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
```

La columna que especifique `ONE_HOT_ENCODING` puede tener un máximo de diez (10) valores distintos.

Parameters

- `sourceColumn`: el nombre de una columna existente. La columna puede tener un máximo de 10 valores distintos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SCALE

Escala o normaliza el rango de datos de una columna numérica.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `strategy`— La operación que se va a aplicar a los valores de las columnas:
  - `MIN_MAX`— Cambia la escala de los valores a un rango de [0,1]
  - `SCALE_BETWEEN`— Cambia la escala de los valores a un rango de 2 valores específicos.
  - `MEAN_NORMALIZATION`— Cambia la escala de los datos para que tengan una media (μ) de 0 y una desviación estándar (σ) de 1 dentro de un rango de [-1, 1]
  - `Z_SCORE`— Escala linealmente los valores de los datos para que tengan una media (μ) de 0 y una desviación estándar (σ) de 1. Ideal para tratar valores atípicos.
- `targetColumn`— El nombre de la columna que contiene los resultados.

Example Ejemplo

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

ASIMETRÍA

Aplica transformaciones a los valores de los datos para cambiar la forma de la distribución y su sesgo.

Parameters

- `sourceColumn`: el nombre de una columna existente.

  `targetColumn`: el nombre de la nueva columna que se va a crear.

  `skewFunction`

- ROOT— extraer la raíz del valor. La raíz se puede proporcionar en el `value` parámetro.

  LOG— valor base logarítmico. La base logarítmica se puede proporcionar en el `value` parámetro.

  SQUARE— función cuadrada

`value`— Argumento de la función skewFunction.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

TOKENIZACIÓN

Divide el texto en unidades más pequeñas, o fichas, como palabras o términos individuales.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `delimiter`— Un delimitador personalizado que aparece entre las palabras tokenizadas. (El comportamiento predeterminado es separar cada ficha por un espacio).
- `expandContractions`— Si`ENABLED`, expande las palabras contraídas. Por ejemplo: «no» se convierte en «no hacer».
- `stemmingMode`— Divide el texto en unidades o símbolos más pequeños, como palabras o términos individuales en minúscula. Hay dos modos de derivación disponibles: |. `PORTER` `LANCASTER`
- `stopWordRemovalMode`— Elimina palabras comunes como a, an, the y más.

- `customStopWords`— Para `StopWordRemovalMode`, permite especificar una lista personalizada de palabras vacías.
- `targetColumn`— El nombre de la columna que contiene los resultados.

Example Ejemplo

```
{
    "Action": {
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

Funciones matemáticas

A continuación, encontrará temas de referencia para funciones matemáticas que funcionan con acciones de receta.

Temas

- ABSOLUTE
- ADD
- CEILING
- DEGREES
- DIVIDIR
- EXPONENTE
- FLOOR
- IS_EVEN

- IS_ODD
- LN
- LOG
- MOD
- MULTIPLICAR
- NEGAR
- PI
- POWER
- RADIANS
- RANDOM
- RANDOM_BETWEEN
- ROUND
- SIGN
- SQUARE_ROOT
- RESTAR

ABSOLUTE

Devuelve el valor absoluto del número introducido en una columna nueva. El valor absoluto indica qué tan lejos está el número de cero, independientemente de si es positivo o negativo

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
```

```
        }
    }
}
```

ADD

Suma los valores de la columna de entrada en una nueva columna, utilizando (`sourceColumn1+sourceColumn2`) o (`sourceColumn1+value1`).

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `value1`— Un valor numérico.
- `sourceColumn2`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

CEILING

Devuelve el número entero más pequeño mayor o igual a los números decimales ingresados en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value1`— Un valor numérico.

- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "CEILING",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

DEGREES

Convierte los radianes de un ángulo en grados y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

DIVIDIR

Divide un número de entrada por otro y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `value1`— Un valor numérico.
- `sourceColumn2`: el nombre de una columna existente.
- `value2`— Un valor numérico.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
        }
    }
}
```

EXPONENTE

Devuelve el número de Euler elevado al enésimo grado en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
```

```
            "targetColumn": "age_EXPONENT"
        }
    }
}
```

FLOOR

Devuelve el número entero más grande mayor o igual al número introducido en una columna nueva.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `value`— Un valor numérico.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

IS_EVEN

Devuelve un valor booleano en una columna nueva que indica si la columna o el valor de origen son pares. Si la columna o el valor de origen es decimal, el resultado es falso.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `trueString`: una cadena que indica si el valor es par.
- `falseString`— Una cadena que indica si el valor no es par.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
        "Parameters": {
            "falseString": "Value is odd",
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

IS_ODD

Devuelve un valor booleano en una columna nueva que indica si la columna o el valor de origen son impares. Si la columna o el valor de origen es decimal, el resultado es falso.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `trueString`— Una cadena que indica si el valor es impar.
- `falseString`— Una cadena que indica si el valor no es impar.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
```

```
}
```

LN

Devuelve el logaritmo natural (número de Euler) de un valor de una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
        }
    }
}
```

LOG

Devuelve el logaritmo de un valor de una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `base`— La base del logaritmo. El valor predeterminado es 10.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "LOG",
```

```
        "Parameters": {
            "base": "10",
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
    }
}
```

MOD

Devuelve el porcentaje de un número respecto a otro número en una columna nueva.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

MULTIPLICAR

Multiplica dos números y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `value1`— Un valor numérico.

- `sourceColumn2`: el nombre de una columna existente.
- `value2`— Un valor numérico.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

NEGAR

Niega un valor y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

Devuelve el valor de pi (3,141592653589793) en una columna nueva.

Parameters

- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

POWER

Devuelve el valor de un número elevado a la potencia del exponente en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`— Un número cuyo valor se va a aumentar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.
- `exponent`— La potencia a la que se elevará el valor.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "POWER",
        "Parameters": {
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
    }
}
```

RADIANS

Convierte los grados en radianes (divide entre 180/pi) y devuelve el valor en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

RANDOM

Devuelve un número aleatorio entre 0 y 1 en una columna nueva.

Parameters

- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "RANDOM",
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

RANDOM_BETWEEN

En una columna nueva, devuelve un número aleatorio entre un límite inferior especificado (incluido) y un límite superior especificado (incluido).

Parameters

- `lowerBound`— El límite inferior del rango de números aleatorios.
- `upperBound`— El límite superior del rango de números aleatorios.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

ROUND

Redondea un valor numérico al entero más cercano de una columna nueva. Se redondea cuando la fracción es 0,5 o más.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

SIGN

Devuelve una nueva columna con -1 si el valor es menor que 0, 0 si el valor es 0 y +1 si el valor es mayor que 0.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

SQUARE_ROOT

Devuelve la raíz cuadrada de un valor de una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

RESTAR

Resta un número de otro y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `value1`— Un valor numérico.
- `sourceColumn2`: el nombre de una columna existente.
- `value2`— Un valor numérico.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "targetColumn": "weight_minus_10_kg",
            "value2": "10"
        }
    }
}
```

Funciones de agregación

A continuación, encontrará temas de referencia para funciones de agregación que funcionan con acciones de receta.

Temas

- ANY
- AVERAGE
- COUNT
- COUNT_DISTINCT
- KTH_LARGEST
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN
- MODE
- DESVIACIÓN_ESTÁNDAR
- SUM
- VARIANCE

ANY

Devuelve los valores de las columnas de origen seleccionadas en una columna nueva. Los valores vacíos y nulos se ignoran.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

AVERAGE

Calcula el promedio de los valores de las columnas de origen y devuelve el resultado en una nueva columna. Se ignora todo lo que no sea un número.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

COUNT

Devuelve el número de valores de las columnas de origen seleccionadas en una columna nueva. Los valores vacíos y nulos se ignoran.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

COUNT_DISTINCT

Devuelve el número total de valores distintos de las columnas de origen seleccionadas en una columna nueva. Los valores vacíos y nulos se ignoran.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "COUNT_DISTINCT",
```

```
        "Parameters": {
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
    }
}
```

KTH_LARGEST

Devuelve el número k más grande de las columnas de origen seleccionadas en una columna nueva.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.
- `value`— Un número que representa k.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

Devuelve el número único más grande de las columnas de origen seleccionadas en una columna nueva.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.

- `targetColumn`: un nombre para la columna recién creada.

  `value`— Un número que representa k.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

Devuelve el valor numérico máximo de las columnas de origen seleccionadas en una nueva columna. Se ignora todo lo que no sea un número.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
}
```

MEDIAN

Devuelve la mediana, el número central de un grupo ordenado de números, de las columnas de origen seleccionadas en una columna nueva. Se omite todo lo que no sea un número.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

Devuelve el valor mínimo de las columnas de origen seleccionadas en una nueva columna. Se ignora todo lo que no sea un número.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MIN",
```

```
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

Devuelve el modo, el número que aparece con más frecuencia, de las columnas de origen seleccionadas en una nueva columna. Se omite todo lo que no sea un número. Para varios modos, el modo se calcula con la función modal.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

DESVIACIÓN_ESTÁNDAR

Devuelve la desviación estándar de las columnas de origen seleccionadas en una columna nueva.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

SUM

Devuelve la suma de los valores de las columnas de origen seleccionadas en una columna nueva. Todo lo que no sea un número se trata como 0.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
    }
}
```

VARIANCE

Devuelve la varianza de las columnas de origen seleccionadas en una nueva columna. La varianza se define como. `Var(X) = [Sum ((X – mean(X))^2)]/Count(X)`

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa una lista de columnas existentes.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "VARIANCE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

Funciones de texto

A continuación, encontrará temas de referencia sobre las funciones de texto que funcionan con acciones de receta.

Temas

- CHAR
- ENDS_WITH
- EXACTO
- ENCONTRAR
- LEFT
- LEN
- LOWER
- COMBINAR_COLUMNAS_Y_VALORES
- APROPIADO
- REMOVE_SYMBOLS
- REMOVE_WHITESPACE
- REPEAT_STRING

- RIGHT
- RIGHT_FIND
- STARTS_WITH
- CADENA_MAYOR_QUE
- STRING_GREATE_THAN_EQUAL
- STRING_LESS_THAN
- STRING_LESS_THAN_EQUAL
- SUBSTRING
- TRIM
- UNICODE
- UPPER

CHAR

Devuelve en una nueva columna el carácter Unicode de cada número entero de la columna de origen o de un valor entero personalizado.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`— Un entero que representa un valor Unicode.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

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

```
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
}
```

ENDS_WITH

`true`Retorna una nueva columna si el número especificado de caracteres situados más a la derecha, o una cadena personalizada, coincide con un patrón.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `pattern`— Una expresión regular que debe coincidir con el final de la cadena.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
```

```
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

EXACTO

Crea una nueva columna que se rellena con uno de los siguientes elementos:

- `True`si una cadena de una columna (o valor) coincide exactamente con otra cadena de una columna (o valor) diferente.
- `False`si no hay ninguna coincidencia.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.
- `value1`: una cadena de caracteres para evaluar.
- `value2`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar solo una de las siguientes combinaciones:
>
> - Ambas`sourceColumn`*N*.
> - Uno de `sourceColumn`*N* y uno de`value`*N*.
> - Ambos`value`*N*.

Example Ejemplo

```
{
```

```
    "RecipeAction": {
        "Operation": "EXACT",
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

ENCONTRAR

Al buscar de izquierda a derecha, busca las cadenas que coinciden con una cadena especificada de la columna de origen o de un valor personalizado y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— Una expresión regular que se va a buscar.
- `position`— La posición del carácter desde el extremo izquierdo de la cadena para empezar.
- `ignoreCase`— Si`true`, ignora las diferencias entre mayúsculas y minúsculas entre letras mayúsculas y minúsculas. Para imponer una concordancia estricta, utilice en su lugar. `false`
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

LEFT

Dado un número de caracteres, toma el número de caracteres situado más a la izquierda de la cadena de la columna de origen o cadena personalizada y devuelve el número especificado de caracteres situados más a la izquierda en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `position`— La posición de los caracteres desde el extremo izquierdo de la cadena para empezar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "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

Devuelve en una nueva columna la longitud de las cadenas de la columna de origen o de las cadenas personalizadas.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
```

```
        }
    }
}
```

LOWER

Convierte todos los caracteres alfabéticos de las cadenas de la columna de origen o las cadenas personalizadas a minúsculas y devuelve el resultado en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
```

```
    }
}
```

COMBINAR_COLUMNAS_Y_VALORES

Concatena las cadenas de las columnas de origen y devuelve el resultado en una nueva columna. Puede insertar un delimitador entre los valores combinados.

Parameters

- `sourceColumns`— Los nombres de dos o más columnas existentes, en JSON-encoded formato.
- `delimiter`: opcional. Uno o más caracteres para colocar entre los dos valores de las columnas de origen.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
    }
}
```

APROPIADO

Convierte todos los caracteres alfabéticos de las cadenas de la columna de origen o los valores personalizados a mayúsculas y minúsculas y devuelve el resultado en una nueva columna.

En el caso correcto, también denominado mayúscula, la primera letra de cada palabra se escribe en mayúscula y el resto de la palabra se transforma en minúscula. Un ejemplo es: El veloz zorro marrón saltó la valla

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_proper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "value": "MR. H. SMITH, ESQ.",
            "targetColumn": "formal_name_proper"
        }
    }
}
```

REMOVE_SYMBOLS

Elimina los caracteres que no sean letras, números, caracteres latinos acentuados o espacios en blanco de las cadenas de la columna de origen o de las cadenas personalizadas y devuelve el resultado en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "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

Elimina los espacios en blanco de las cadenas de la columna de origen o de las cadenas personalizadas y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "sourceColumn": "job_desc",
            "targetColumn": "job_desc_remove_whitespace"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "value": "This string has spaces in it",
            "targetColumn": "string_without_spaces"
        }
    }
}
```

REPEAT_STRING

Repite las cadenas de la columna de origen o del valor de entrada personalizado un número específico de veces y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `count`— El número de veces que se va a repetir la cadena.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "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 un número de caracteres, toma el número de caracteres situado más a la derecha de las cadenas de la columna de origen o de las cadenas personalizadas y devuelve el número especificado de caracteres situados más a la derecha en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `position`— La posición de los caracteres desde el lado derecho de la cadena para empezar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "RIGHT",
        "Parameters": {
            "sourceColumn": "nationality",
            "position": "3",
            "targetColumn": "nationality_right"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "RIGHT",
        "Parameters": {
            "value": "United States of America",
            "position": "7",
            "targetColumn": "usa_right"
        }
    }
}
```

RIGHT_FIND

Al buscar de derecha a izquierda, busca las cadenas que coinciden con una cadena especificada de la columna de origen o de un valor personalizado y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `pattern`— Una expresión regular para buscar.
- `position`— La posición del carácter desde el extremo derecho de la cadena para empezar.
- `ignoreCase`— Si`true`, ignora las diferencias entre mayúsculas y minúsculas entre letras mayúsculas y minúsculas. Para imponer una concordancia estricta, utilice en su lugar. `false`
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

Devuelve `true` una nueva columna si el número especificado de caracteres situados más a la izquierda, o una cadena personalizada, coincide con un patrón.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `pattern`— Una expresión regular que debe coincidir con el inicio de la cadena.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STARTS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
        }
    }
}
```

CADENA_MAYOR_QUE

Crea una nueva columna que se rellena con uno de los siguientes elementos:

- `True`si una cadena de una columna (o valor) es mayor que otra cadena de una columna (o valor) diferente.
- `False`si no hay ninguna coincidencia.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.
- `value1`: una cadena de caracteres para evaluar.
- `value2`: una cadena de caracteres para evaluar.

- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar solo una de las siguientes combinaciones:
>
> - Ambas`sourceColumn`*N*.
> - Uno de `sourceColumn`*N* y uno de`value`*N*.
> - Ambos`value`*N*.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
        }
    }
}
```

STRING_GREATE_THAN_EQUAL

Crea una nueva columna que se rellena con uno de los siguientes elementos:

- `True`si una cadena de una columna (o valor) es mayor o igual que otra cadena de una columna (o valor) diferente.
- `False`si no hay ninguna coincidencia.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.
- `value1`: una cadena de caracteres para evaluar.

- `value2`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar solo una de las siguientes combinaciones:
>
> - Ambas`sourceColumn`*N*.
> - Uno de `sourceColumn`*N* y uno de`value`*N*.
> - Ambos`value`*N*.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "nationality",
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

STRING_LESS_THAN

Crea una nueva columna que se rellena con uno de los siguientes elementos:

- `True`si una cadena de una columna (o valor) es menor que otra cadena de una columna (o valor) diferente.
- `False`si no hay ninguna coincidencia.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.

- `value1`: una cadena de caracteres para evaluar.
- `value2`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar solo una de las siguientes combinaciones:
>
> - Ambas`sourceColumn`*N*.
> - Uno de `sourceColumn`*N* y uno de`value`*N*.
> - Ambos`value`*N*.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

STRING_LESS_THAN_EQUAL

Crea una nueva columna que se rellena con uno de los siguientes elementos:

- `True`si una cadena de una columna (o valor) es menor o igual que otra cadena de una columna (o valor) diferente.
- `False`si no hay ninguna coincidencia.

Parameters

- `sourceColumn1`: el nombre de una columna existente.

- `sourceColumn2`: el nombre de una columna existente.
- `value1`: una cadena de caracteres para evaluar.
- `value2`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar solo una de las siguientes combinaciones:
>
> - Ambas`sourceColumn`*N*.
> - Uno de `sourceColumn`*N* y uno de`value`*N*.
> - Ambos`value`*N*.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

Devuelve en una nueva columna algunas o todas las cadenas especificadas en la columna de origen, en función de los valores de índice inicial y final definidos por el usuario.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `startPosition`— La posición de los caracteres para empezar, desde el extremo izquierdo de la cadena.

- `endPosition`— La posición del carácter con la que termina, desde el extremo izquierdo de la cadena.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
}
```

TRIM

Elimina los espacios en blanco iniciales y finales de las cadenas de la columna de origen o de las cadenas personalizadas y devuelve el resultado en una columna nueva. Los espacios entre las palabras no se eliminan.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
    }
}
```

UNICODE

Devuelve en una nueva columna el valor del índice Unicode del primer carácter de las cadenas de la columna de origen o de las cadenas personalizadas.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
        }
    }
}
```

UPPER

Convierte todos los caracteres alfabéticos de las cadenas de la columna de origen o las cadenas personalizadas a mayúsculas y devuelve el resultado en una columna nueva.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
        }
    }
}
```

Funciones de fecha y hora

A continuación, encontrará temas de referencia para las funciones de fecha y hora que funcionan con acciones de recetas.

Temas

- CONVERT_TIMEZONE
- DATE
- DATE_ADD

- DATE_DIFF
- DATE_FORMAT
- DATE_TIME
- DAY
- HOUR
- MILLISECOND
- MINUTE
- MONTH
- NOMBRE_MES
- NOW
- CUARTO
- SECOND
- TIME
- HOY
- UNIX_TIME
- UNIX_TIME_FORMAT
- DÍA_SEMANA
- NÚMERO_SEMANA
- YEAR

CONVERT_TIMEZONE

Convierte un valor de hora de la columna de origen en una nueva columna en función de una zona horaria específica.

Parameters

- `sourceColumn`: el nombre de una columna existente. La columna de origen puede ser del tipo `stringdate`, o. `timestamp`
- `fromTimeZone`— Zona horaria del valor de origen. Si no se especifica nada, la zona horaria predeterminada es UTC.

- `toTimeZone`— Zona horaria a la que se va a convertir. Si no se especifica nada, la zona horaria predeterminada es UTC.
- `targetColumn`— Un nombre para la columna recién creada.
- `dateTimeFormat`: opcional. Cadena de formato para la fecha. Si no se especifica el formato, se usa el formato predeterminado:`yyyy-mm-dd HH:MM:SS`.

Example Ejemplo

```
{
    "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

Crea una nueva columna que contiene el valor de la fecha, de las columnas de origen o de los valores proporcionados.

Parameters

- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica esta cadena, el formato predeterminado es`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Una JSON-encoded cadena que representa los componentes de la fecha y la hora:
  - `year`
  - `value`
  - `month`
  - `day`

- `hour`
- `second`

Cada componente debe especificar uno de los siguientes elementos:

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.

- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "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

Añade un año, un mes o un día a la fecha desde una columna o un valor de origen y crea una nueva columna que contiene los resultados.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `units`— Una unidad de medida para ajustar la fecha. Los valores válidos son `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`, `WEEKSSECONDS`,`DAYS`, y`MINUTES`.
- `dateAddValue`— El número de `units` que se va a añadir a la fecha.
- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica, el formato predeterminado es `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
}
```

DATE_DIFF

Crea una nueva columna que contiene la diferencia entre dos fechas.

Parameters

- `sourceColumn1`: el nombre de una columna existente.
- `sourceColumn2`: el nombre de una columna existente.
- `value1`: una cadena de caracteres para evaluar.
- `value2`: una cadena de caracteres para evaluar.
- `units`— Una unidad de medida para describir la diferencia entre las fechas. Los valores válidos son `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`, `WEEKSSECONDS`,`DAYS`, y`MINUTES`.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Solo puede especificar una de las siguientes combinaciones:

- Ambas de `sourceColumn1` y`sourceColumn2`.
- Uno de `sourceColumn1` o `sourceColumn2` y uno de `value1` o`value2`.
- Ambos de `value1` y`value2`.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
        }
    }
}
```

DATE_FORMAT

Crea una nueva columna que contiene una fecha, en un formato específico, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`— Una cadena para evaluar.
- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica, el formato predeterminado es `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplos

```
{
    "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

Crea una nueva columna que contiene el valor de fecha y hora, a partir de las columnas de origen o de los valores proporcionados.

Parameters

- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica esta cadena, el formato predeterminado es`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Una JSON-encoded cadena que representa los componentes de la fecha y la hora:
  - `year`
  - `value`
  - `month`

- `day`
- `hour`
- `second`

Cada componente debe especificar uno de los siguientes elementos:

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.

Example Ejemplo

```
{
    "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

Crea una nueva columna que contiene el día del mes, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

HOUR

Crea una nueva columna que contiene el valor de la hora, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
```

```
}
```

MILLISECOND

Crea una nueva columna que contiene el valor de milisegundos de una columna de origen o un valor de entrada.

Parameters

- `sourceColumn`: el nombre de una columna existente. La columna de origen puede ser del tipo`string`,`date`, o`timestamp`.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`— Un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
}
```

MINUTE

Crea una nueva columna que contiene el valor de los minutos, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.

- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
}
```

MONTH

Crea una nueva columna que contiene el número del mes, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

NOMBRE_MES

Crea una nueva columna que contiene el nombre del mes, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

NOW

Crea una nueva columna que contiene la fecha y la hora actuales en el formatoyyyy-mm-dd HH:MM:SS.

Parameters

- timeZone— El nombre de una zona horaria. Si no se especifica ninguna zona horaria, el valor predeterminado es la hora universal coordinada (UTC).
- targetColumn: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
}
```

CUARTO

Crea una nueva columna que contiene el trimestre basado en fechas a partir de una cadena que representa una fecha.

> **Note**
>
> Los trimestres se designan en la nueva columna como 1, 2, 3 o 4.
>
> - El 1 corresponde a enero, febrero y marzo.
> - El 2 es abril, mayo y junio.
> - El 3 es julio, agosto y septiembre.
> - El 4 es octubre, noviembre y diciembre.

Parameters

- `sourceColumn`: el nombre de una columna existente. La columna de origen puede ser del tipo `stringdate`, `otimestamp`.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`— Un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
}
```

SECOND

Crea una nueva columna que contiene el segundo valor, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

Crea una nueva columna que contiene el valor de la hora, a partir de las columnas o valores de origen proporcionados.

Parameters

- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica esta cadena, el formato predeterminado es`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Una JSON-encoded cadena que representa los componentes de la fecha y la hora:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Cada componente debe especificar uno de los siguientes elementos:

  - `sourceColumn`: el nombre de una columna existente.
  - `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

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

HOY

Crea una nueva columna que contiene la fecha actual en el formatoyyyy-mm-dd.

Parameters

- `timeZone`— El nombre de una zona horaria. Si no se especifica ninguna zona horaria, el valor predeterminado es la hora universal coordinada (UTC).
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
```

```
        }
    }
}
```

UNIX_TIME

Crea una nueva columna que contiene un número que representa la época (hora de Unix) (el número de segundos transcurridos desde el 1 de enero de 1970) en función de una columna de origen o un valor de entrada. Si se puede deducir la zona horaria, la salida se encuentra en esa zona horaria. De lo contrario, la salida está en hora universal coordinada (UTC).

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
        }
    }
}
```

UNIX_TIME_FORMAT

Convierte la hora de Unix de una columna de origen o un valor de entrada a un formato de fecha numérico especificado y devuelve el resultado en una nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`— Un número entero que representa una marca temporal de una época de Unix.
- `dateTimeFormat`: opcional. Una cadena de formato para la fecha, tal como aparecerá en la nueva columna. Si no se especifica, el formato predeterminado es `yyyy-mm-dd HH:MM:SS`.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

DÍA_SEMANA

Crea una nueva columna que contiene el día de la semana a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

NÚMERO_SEMANA

Crea una nueva columna que contiene el número de la semana (del 1 al 52), a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
```

```
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

YEAR

Crea una nueva columna que contiene el año, a partir de una cadena que representa una fecha.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: un nombre para la columna recién creada.

> **Note**
>
> Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
}
```

Funciones de ventana

A continuación, encontrará temas de referencia para las funciones de ventana que funcionan con acciones de receta.

Temas

FILL

Devuelve una nueva columna basada en una columna de origen especificada. Para cualquier valor nulo o que falte en la columna de origen, `FILL` elige el valor no vacío más reciente de una ventana de filas antes y después del valor de origen en cuestión. A continuación, el valor elegido se coloca en la nueva columna.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representan el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
}
```

NEXT

Devuelve una columna nueva, donde cada valor representa un valor que está n filas más adelante en la columna de origen.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- numRows— Un valor que representa n filas anteriores en la columna de origen. Por ejemplo, si numRows es 3, NEXT utiliza el tercer `sourceColumn` valor siguiente como `targetColumn` valor nuevo.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
}
```

ANTERIOR

Devuelve una nueva columna, donde cada valor representa un valor que está n filas antes en la columna de origen.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRows`— Un valor que representa n filas anteriores en la columna de origen. Por ejemplo, si `numRows` es 3, `PREV` utiliza el tercer `sourceColumn` valor anterior como `targetColumn` valor nuevo.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

PROMEDIO_ACUMULABLE

Devuelve, en una nueva columna, el promedio móvil de los valores desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.

- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
    }
}
```

ROLLING_COUNT_A

Devuelve en una nueva columna el recuento continuo de valores no nulos desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
```

```
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

Devuelve, en una nueva columna, el késimo valor más alto acumulado desde un número específico de filas anteriores hasta un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `value`— El valor de k.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
  "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

Devuelve, en una nueva columna, el k ésimo valor más alto de un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `value`— El valor de k.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
  "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

Devuelve en una nueva columna el número máximo continuo de valores desde un número especificado de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.

numRowsBefore— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.

- numRowsAfter— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- targetColumn: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_MAX",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

ROLLING_MIN

Devuelve en una nueva columna el mínimo continuo de valores desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- sourceColumn: el nombre de una columna existente.

  numRowsBefore— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- numRowsAfter— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- targetColumn: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

MODO RODANTE

Devuelve en una nueva columna el modo de rotación (el valor más común) desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- ModeType — La función modal que se va a aplicar a la ventana. Los valores válidos son `NONE`, `MINIMUM`, `MAXIMUM` y `AVERAGE`.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
```

```
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

DESVIACIÓN ESTÁNDAR RODANTE

Devuelve en una nueva columna la desviación estándar continua de los valores desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- numRowsBefore— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- numRowsAfter— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

ROLLING_SUM

Devuelve en una nueva columna la suma continua de los valores desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.

  `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

ROLLING_VARIANCE

Devuelve en una nueva columna la varianza continua de los valores desde un número específico de filas anteriores a un número específico de filas después de la fila actual de la columna especificada.

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `numRowsBefore`— Un número de filas antes de la fila de origen actual, que representa el inicio de la ventana.
- `numRowsAfter`— Un número de filas después de la fila de origen actual, que representan el final de la ventana.
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

Devuelve en una nueva columna un identificador de sesión basado en una ventana creada por los nombres de las columnas de las instrucciones «agrupar por» y «ordenar por».

Parameters

- `groupByColumns`— Una JSON-encoded cadena que describe las columnas «agrupar por».
- `orderByColumns`— Una JSON-encoded cadena que describe las columnas «ordenadas por».
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESSION

Devuelve en una nueva columna un identificador de sesión basado en una ventana creada con los nombres de las columnas de las instrucciones «agrupar por» y «ordenar por».

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `units`— Una unidad de medida para describir la duración de la sesión. Los valores válidos son `MONTHS` `YEARSMILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`, `WEEKSSECONDS`,`DAYS`, y`MINUTES`.
- `value`— El número de `units` para definir el período de tiempo.
- `groupByColumns`— Una JSON-encoded cadena que describe las columnas «agrupar por».
- `orderByColumns`— Una JSON-encoded cadena que describe las columnas «ordenadas por».
- `targetColumn`: un nombre para la columna recién creada.

Example Ejemplo

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
}
```

Funciones web

A continuación, encontrará temas de referencia sobre las funciones web que funcionan con acciones de recetas.

Temas

IP_TO_INT

Convierte el valor del Protocolo de Internet versión 4 (IPv4) de la columna de origen u otro valor en el valor entero correspondiente de la columna de destino y devuelve el resultado en una nueva columna. Esta función solo funciona para IPv4.

Por ejemplo, considere la siguiente dirección IP.

```
192.168.1.1
```

Si utiliza este valor como entrada para`IP_TO_INT`, el valor de salida es el siguiente.

```
3232235777
```

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_TO_IP

Convierte el valor entero de la columna de origen u otro valor en el valor IPv4 correspondiente de la columna de destino y devuelve el resultado en una nueva columna. Esta función solo funciona para IPv4.

Por ejemplo, considere el siguiente entero.

```
167772410
```

Si utiliza este valor como entrada para`INT_TO_IP`, el valor de salida es el siguiente.

```
10.0.0.250
```

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
[  {
    "RecipeAction": {
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
  }
]
```

URL_PARAMS

Extrae los parámetros de consulta de una cadena URL, los formatea como un objeto JSON y devuelve el resultado en una nueva columna.

Por ejemplo, considere la siguiente URL.

```
https://example.com/?firstParam=answer&secondParam=42
```

Si utiliza este valor como entrada para`URL_PARAMS`, el valor de salida es el siguiente.

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parameters

- `sourceColumn`: el nombre de una columna existente.
- `value`: una cadena de caracteres para evaluar.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Puede especificar `sourceColumn` o `value`, pero no ambos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
    }
}
```

Otras funciones

A continuación, encontrará temas de referencia para otras funciones que funcionan con acciones de recetas.

Temas

- COALESCE
- GET_ACTION_RESULT
- GET_STEP_DATAFRAME

COALESCE

Devuelve en una nueva columna el primer valor no nulo encontrado en la matriz de columnas. El orden de las columnas enumeradas en la función determina el orden en que se buscan.

Parameters

- `sourceColumns`— Una JSON-encoded cadena que representa la lista de columnas existentes.
- `targetColumn`: el nombre de la nueva columna que se va a crear.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

Obtiene el resultado de una acción enviada anteriormente. Solo para su uso en la experiencia interactiva.

Parameters

- `actionId`— El ActionId devuelto en la SendProjectSessionAction respuesta original.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "GET_ACTION_RESULT",
        "Parameters": {
            "actionId": "7",
```

```
        }
    }
}
```

GET_STEP_DATAFRAME

Obtiene el marco de datos de un paso de la receta del proyecto. Solo para su uso en la experiencia interactiva. Se utiliza con el ViewFrame parámetro para paginar en un marco de datos grande.

Parameters

- `stepIndex`— El índice del paso de la receta del proyecto para obtener el marco de datos.

Example Ejemplo

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

Cuotas para AWS Glue DataBrew

Puede ver sus cuotas DataBrew de servicio en la consola AWS Service Quotas. También puedes solicitar un aumento de cuota, para cualquier cuota que sea ajustable.

Historial de documentos para AWS Glue DataBrew Guía para desarrolladores

Versión actual de la API: databrew-2017-07-25

En la siguiente tabla se describe la documentación de esta versión de.AWS Glue DataBrew Si desea recibir una notificación cuando se actualice la Guía para AWS Glue DataBrew desarrolladores, puede suscribirse a la fuente RSS.

Cambio	Descripción	Fecha
`glue:GetCustomEntityType` agregado a las políticas AWS administradas	Este permiso es necesario para ejecutar trabajos AWS Glue DataBrew de perfil si PII-identification está activado. Para obtener más información, consulte AWS Glue DataBrew las actualizaciones de las políticas AWS gestionadas.	20 de marzo de 2024
Support para múltiples algoritmos de hash en la transformación CRYPTOGRAPHIC_HASH	Ahora puede especificar un algoritmo de hash al aplicar hash a los valores de una columna. Para obtener más información, consulte CRYPTOGRAPHIC_HASH.	11 de agosto de 2023
`glue:BatchGetCustomEntityTypes` agregado a las políticas administradas AWS	Este permiso es necesario para ejecutar trabajos AWS Glue DataBrew de perfil si PII-identification está activado. Para obtener más información, consulte AWS Glue DataBrew las actualizaciones de las políticas AWS gestionadas.	9 de mayo de 2022

Soporte para el formato de archivo ORC Apache	DataBrew ahora es compatible con Apache ORC como formato de archivo para fuentes y DataBrew salidas de datos. Para obtener más información, consulte Tipos de archivos compatibles para las fuentes de datos.	31 de marzo de 2022
Support para el acceso multicuenta a AWS Glue Data Catalog Amazon S3	Ahora puede acceder a las tablas de AWS Glue Data Catalog S3 desde otras tablas Cuentas de AWS si se ha creado una política de recursos adecuada en la AWS Glue consola. Tras crear una política, las tablas S3 del catálogo de datos correspondientes se pueden seleccionar como fuentes de entrada al crear un DataBrew conjunto de datos. Para obtener más información, consulte Conexiones compatibles para fuentes y salidas de datos.	11 de marzo de 2022

Support para la integración de consolas nativas con Amazon AppFlow	DataBrew ahora tiene una integración de consola nativa con Amazon AppFlow. Esta integración significa que puede conectarse a los datos de Salesforce, Zendesk, Slack y otras aplicaciones de software ServiceNow como servicio (SaaS). También puede conectarse a datos de Servicios de AWS Amazon S3 y Amazon Redshift. Para obtener más informaci ón, consulte Conexiones compatibles para fuentes y salidas de datos.	18 de noviembre de 2021
Support for data quality rules	DataBrew ahora admite la creación de reglas de calidad de los datos, que son comprobaciones de validació n personalizables que definen los requisitos empresariales para datos específicos. Para obtener más información, consulte Validación de la calidad de los datos en AWS Glue DataBrew.	18 de noviembre de 2021

Support para sentencias SQL personalizadas	DataBrew ahora admite sentencias SQL personalizadas para recuperar datos de Amazon Redshift y Snowflake. Esta compatibilidad significa que puede usar una consulta diseñada específicamente para seleccionar y limitar los datos devueltos por tablas grandes. Para obtener más información, consulte Conexiones compatibles para fuentes y salidas de datos.	18 de noviembre de 2021
Support for PII detection	DataBrew ahora admite la detección de información de identificación personal (PII). Esto le da la opción de enmascarar la PII durante la preparación de los datos. Para obtener más información, consulte Identificación y manejo de la información de identificación personal (PII).	18 de noviembre de 2021
Support para AWS regiones adicionales	DataBrew ahora admite AWS regiones adicionales. Para ver una lista de las regiones compatibles, consulta los AWS Glue DataBrew puntos finales y las cuotas.	5 de octubre de 2021

Support para escribir datos en tablas Formation-based Amazon S3 de Lake	DataBrew ahora admite la escritura de datos en tablas de AWS Glue Data Catalog S3 en función de AWS Lake Formation. DataBrew ahora también admite la escritura de datos en el formato Tableau Hyper. Para obtener más información, consulte Crear trabajos de preparación y trabajar con AWS Glue DataBrew ellos.	13 de agosto de 2021
Support para escribir datos en destinos JDBC	DataBrew ahora admite la escritura de datos directame nte en JDBC-supported bases de datos y almacenes de datos. Estos incluyen Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database y PostgreSQL. Para obtener más informaci ón, consulte Crear trabajos de recetas y trabajar con ellos.AWS Glue DataBrew	23 de julio de 2021
Support para especificar qué estadísticas de calidad de datos se generan para un trabajo de perfil	DataBrew ahora permite especificar qué estadísticas de calidad de datos se generan automáticamente para los conjuntos de datos de un trabajo de perfil. Para obtener más información, consulte Crear trabajos de preparación y trabajar con ellos AWS Glue DataBrew.	23 de julio de 2021

Support para escribir conjuntos de datos en el AWS Glue Data Catalog	DataBrew ahora incluye soporte para escribir conjuntos de datos directamente en el.AWS Glue Data Catalog Puede optar por almacenar los conjuntos de datos creados a partir de trabajos que ejecutan sus recetas de preparación de datos en las tablas de Amazon S3, Amazon Redshift y Amazon RDS del catálogo de datos. Las tablas de RDS compatibles incluyen las de Amazon Aurora, RDS de Oracle, RDS de Microsoft SQL Server, RDS de MySQL y RDS de PostgreSQL.	30 de junio de 2021
Support para identificar tipos de datos avanzados	DataBrew ahora incluye soporte para identificar y marcar automáticamente los tipos de datos avanzados para las columnas, lo que facilita la normalización de las columnas que contienen ciertos tipos de datos. Estos tipos de datos incluyen el número de seguro social, la dirección de correo electrónico, el número de teléfono, el sexo, la tarjeta de crédito, la URL, la dirección IP, la fecha y la hora, la moneda, el código postal, el país, la región, el estado y la ciudad.	30 de junio de 2021

Support para usar Amazon AppFlow para transferir datos desde aplicaciones SAAS	DataBrew ahora admite el uso de Amazon AppFlow para transferir datos a Amazon S3 desde aplicaciones de software como servicio (SaaS) de terceros, como Salesforce, Zendesk, Slack y. ServiceNow Para obtener más información, consulte Conexiones compatibles para fuentes y salidas de datos.	29 de abril de 2021
Support para crear DataBrew conjuntos de datos con entradas de bases de datos JDBC	DataBrew ahora admite la creación de conjuntos de datos a partir de datos de JDBC-supported bases de datos y almacenes de datos, incluidos Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database y PostgreSQL. Para obtener más información, consulte Conexiones compatibles para fuentes y salidas de datos.	2 de abril de 2021
Support para más Regiones de AWS	DataBrew ahora admite más Regiones de AWS. Para ver una lista de las regiones compatibles, consulta los AWS Glue DataBrew puntos finales y las cuotas.	28 de enero de 2021

Nuevas transformaciones para gestionar la duplicación	Se han agregado cuatro nuevas transformaciones a la DataBrew consola y a la API para gestionar la duplicación. Para obtener más información, consulte DELETE_DUPLICATE_ROWS, FLAG_DUPLICATE_ROWS, FLAG_DUPLICATES_IN_COLUMNy REMOVE_DUPLICATES en los pasos de la receta de calidad de los datos.	28 de enero de 2021
Delimitadores CSV adicionales	DataBrew ahora admite delimitadores adicionales además de las comas en los archivos de valores separados por comas (CSV) que se utilizan para crear conjuntos de datos. DataBrew Para obtener más información, consulte Creación y uso de conjuntos de datos.AWS Glue DataBrew	28 de enero de 2021
DataBrew extensión para JupyterLab	Ahora se puede utilizar AWS Glue DataBrew como extensión en JupyterLab. Para obtener más información, consulte DataBrew Utilización como extensión en JupyterLab.	20 de noviembre de 2020
Nueva herramienta de preparación de datos:AWS Glue DataBrew	Esta es la primera versión de la Guía para desarrolladores de AWS Glue DataBrew.	11 de noviembre de 2020

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el AWS glosario de la Glosario de AWS Referencia.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.