Unable to locate subtitle

AWS Glue DataBrew Guide du développeur

AWS Glue DataBrew Guide du développeur: ***Unable to locate subtitle***

Table of Contents

Qu'est-ce que AWS Glue DataBrew?

AWS Glue DataBrew est un outil visuel de préparation des données qui permet aux utilisateurs de nettoyer et de normaliser les données sans écrire de code. L'utilisation DataBrew permet de réduire le temps nécessaire à la préparation des données pour l'analyse et l'apprentissage automatique (ML) jusqu'à 80 %, par rapport à une préparation de données développée sur mesure. Vous pouvez choisir parmi plus de 250 transformations prêtes à l'emploi pour automatiser les tâches de préparation des données, telles que le filtrage des anomalies, la conversion des données dans des formats standard et la correction de valeurs non valides.

Grâce à DataBrew cela, les analystes commerciaux, les scientifiques des données et les ingénieurs de données peuvent collaborer plus facilement pour obtenir des informations à partir de données brutes. Grâce DataBrew à la technologie sans serveur, quel que soit votre niveau technique, vous pouvez explorer et transformer des téraoctets de données brutes sans avoir à créer de clusters ni à gérer d'infrastructure.

Grâce à l' DataBrew interface intuitive, vous pouvez découvrir, visualiser, nettoyer et transformer les données brutes de manière interactive. DataBrew fait des suggestions intelligentes pour vous aider à identifier les problèmes de qualité des données qui peuvent être difficiles à détecter et fastidieux à résoudre. En DataBrew préparant vos données, vous pouvez utiliser votre temps pour agir sur les résultats et itérer plus rapidement. Vous pouvez enregistrer la transformation sous forme d'étapes dans une recette, que vous pouvez mettre à jour ou réutiliser ultérieurement avec d'autres ensembles de données, et déployer de manière continue.

L'image suivante montre comment DataBrew fonctionne à un niveau élevé.

Pour l'utiliser DataBrew, vous créez un projet et vous connectez à vos données. Dans l'espace de travail du projet, vos données sont affichées dans une interface visuelle semblable à une grille. Ici, vous pouvez explorer les données et consulter les distributions de valeurs et les graphiques pour comprendre leur profil.

Pour préparer les données, vous pouvez choisir parmi plus de 250 transformations pointer-cliquer. Il s'agit notamment de supprimer les valeurs nulles, de remplacer les valeurs manquantes, de corriger les incohérences du schéma, de créer des colonnes basées sur des fonctions, etc. Vous pouvez également utiliser des transformations pour appliquer des techniques de traitement du langage naturel (NLP) afin de diviser des phrases en phrases. Les aperçus immédiats montrent une partie de vos données avant et après la transformation, afin que vous puissiez modifier votre recette avant de l'appliquer à l'ensemble de données.

Après DataBrew avoir exécuté votre recette sur votre ensemble de données, la sortie est stockée dans Amazon Simple Storage Service (Amazon S3). Une fois que votre ensemble de données nettoyé et préparé est dans Amazon S3, un autre de vos systèmes de stockage ou de gestion des données peut l'ingérer.

Concepts et termes fondamentaux dans AWS Glue DataBrew

Vous trouverez ci-dessous un aperçu des concepts fondamentaux et de la terminologie dans AWS Glue DataBrew. Après avoir lu cette section, consultezDémarrage avec AWS Glue DataBrew,

qui vous explique le processus de création de projets, de connexion d'ensembles de données et d'exécution de tâches.

Rubriques

Project

L'espace de travail interactif de préparation des données DataBrew s'appelle un projet. À l'aide d'un projet de données, vous gérez un ensemble d'éléments connexes : données, transformations et processus planifiés. Dans le cadre de la création d'un projet, vous choisissez ou créez un jeu de données sur lequel travailler. Ensuite, vous créez une recette, qui est un ensemble d'instructions ou d'étapes que vous DataBrew souhaitez suivre. Ces actions transforment vos données brutes en un formulaire prêt à être utilisé par votre pipeline de données.

Jeu de données

Un ensemble de données désigne simplement un ensemble de données, c'est-à-dire des lignes ou des enregistrements divisés en colonnes ou en champs. Lorsque vous créez un DataBrew projet, vous vous connectez aux données que vous souhaitez transformer ou préparer ou les télécharger. DataBrew peut fonctionner avec des données provenant de n'importe quelle source, importées à partir de fichiers formatés, et il se connecte directement à une liste croissante de magasins de données.

En effet DataBrew, un ensemble de données est une connexion en lecture seule à vos données. DataBrew collecte un ensemble de métadonnées descriptives pour faire référence aux données. Aucune donnée réelle ne peut être modifiée ou stockée par DataBrew. Pour des raisons de simplicité, nous utilisons un ensemble de données pour faire référence à la fois à l'ensemble de données réel et aux DataBrew utilisations des métadonnées.

Formule

Dans DataBrew, une recette est un ensemble d'instructions ou d'étapes relatives aux données sur lesquelles vous DataBrew souhaitez agir. Une recette peut contenir de nombreuses étapes, et chaque étape peut contenir de nombreuses actions. Vous utilisez les outils de transformation de la barre d'outils pour configurer toutes les modifications que vous souhaitez apporter à vos données. Plus tard, lorsque vous êtes prêt à voir le produit fini de votre recette, vous lui attribuez cette tâche DataBrew et vous la planifiez. DataBrew stocke les instructions relatives à la transformation des données, mais ne stocke aucune de vos données réelles. Vous pouvez télécharger et réutiliser des recettes dans d'autres projets. Vous pouvez également publier plusieurs versions d'une recette.

Tâche

DataBrew se charge de transformer vos données en exécutant les instructions que vous avez définies lorsque vous avez créé une recette. Le processus d'exécution de ces instructions s'appelle une tâche. Une tâche peut mettre en œuvre vos recettes de données selon un calendrier prédéfini. Mais vous n'êtes pas limité à un calendrier. Vous pouvez également exécuter des tâches à la demande. Si vous souhaitez profiler certaines données, vous n'avez pas besoin d'une recette. Dans ce cas, vous pouvez simplement configurer une tâche de profilage pour créer un profil de données.

Traçabilité des données

DataBrew suit vos données dans une interface visuelle afin de déterminer leur origine, appelée lignée de données. Cette vue vous montre comment les données circulent entre les différentes entités d'où elles proviennent à l'origine. Vous pouvez voir son origine, les autres entités qui l'ont influencée, ce qui lui est arrivé au fil du temps et où il a été stocké.

Profil de données

Lorsque vous profilez vos données, vous DataBrew créez un rapport appelé profil de données. Ce résumé vous renseigne sur la forme actuelle de vos données, notamment le contexte du contenu, la structure des données et leurs relations. Vous pouvez créer un profil de données pour n'importe quel ensemble de données en exécutant une tâche de profilage de données.

Intégrations de produits et services

Utilisez cette section pour savoir à quels produits et services s'intègrent DataBrew.

DataBrew fonctionne avec les AWS services suivants pour la mise en réseau, la gestion et la gouvernance :

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

DataBrew fonctionne avec les lacs de AWS données et les magasins de données suivants :

- AWS Lake Formation
- Amazon S3

DataBrew prend en charge les formats de fichier et les extensions suivants pour le téléchargement de données.

Format	Extension de fichier (facultatif)	Extensions pour les fichiers compressés (obligatoire)

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

classeur Microsoft Excel	`.xlsx`	Aucun support de compressi on

JSON (document JSON et lignes JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`

Format	Extension de fichier (facultatif)	Extensions pour les fichiers compressés (obligatoire)
		`.bz2` `.deflate`
Apache ORC	`.orc`	`.zlib` `.snappy`

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

DataBrew écrit des fichiers de sortie sur Amazon S3 et prend en charge les formats de fichiers et extensions suivants.

Format	Extension de fichier (non compressée)	Extensions de fichiers (compressées)
Comma-separated valeurs	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated valeurs	`.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	Non pris en charge	`.glue.parquet.snappy`

Format	Extension de fichier (non compressée)	Extensions de fichiers (compressées)
Apache Avro	`.avro`	`.avro.snappy` , `.avro.gz`, `.avro.lz4` , `.avro.bz2` , `.avro.deflate` , `.avro.br`
Apache ORC	`.orc`	`.orc.snappy` , `.orc.lzo`, `.orc.zlib`
xml	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`
JSON (format de lignes JSON uniquement)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	Non pris en charge	Non applicable

Configuration AWS Glue DataBrew

Avant de commencer AWS Glue DataBrew, vous devez configurer certaines autorisations, un utilisateur et un rôle. Commencez par suivre les étapes suivantes :

1. Création d'un AWS compte selon les besoins et création de politiques Gestion des identités et des accès AWS(IAM) pour permettre aux utilisateurs d'exécuter DataBrew :
   - Création d'un nouveau AWS compte et ajout d'un utilisateur. Pour de plus amples informations, veuillez consulter Configuration d'un nouveau AWS compte.
   - Ajouter une politique IAM pour un utilisateur de console. Un utilisateur disposant de ces autorisations peut accéder DataBrew au AWS Management Console.
   - Ajouter des autorisations pour les ressources de données pour un rôle IAM. Un rôle IAM doté de ces autorisations peut accéder aux données au nom de l'utilisateur.

   Vous devez être un administrateur IAM pour créer des utilisateurs, des rôles et des politiques.
2. Ajouter des utilisateurs ou des groupes pour DataBrew. Un utilisateur ou un groupe doté des autorisations appropriées peut accéder DataBrew à la console.
3. Ajouter un rôle avec des autorisations d'accès aux données pour DataBrew. Un rôle doté des autorisations appropriées peut accéder aux données au nom de l'utilisateur.

Configuration d'un nouveau AWS compte

Si vous n'avez pas de AWS compte, créez-en un AWS et créez un utilisateur administrateur IAM.

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrir https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d’inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l’utilisateur racine a accès à l’ensemble des Services AWS et des ressources

de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

Afin de créer un utilisateur administrateur, choisissez l'une des options suivantes :

Choisissez un moyen de gérer votre administrateur	À	En	Vous pouvez également
Dans IAM Identity Center (Recommandé)	Utiliser des informations d'identification à court terme pour accéder àAWS. C'est conforme aux bonnes pratiques en matière de sécurité. Pour plus d'informations sur les bonnes pratiques, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.	Suivant les instructions fournies dans Mise en route dans le Guide de l'utilisateur AWS IAM Identity Center.	Configurez l'accès par programmation en configurant le AWS CLIà utiliser AWS IAM Identity Center dans le guide de l'AWS Command Line Interface utilisateur.
Dans IAM (Non recommandé)	Utiliser les informations d'identification à long terme pour accéder àAWS.	Suivant les instructions fournies dans Création d'un utilisateur IAM pour l'accès d'urgence dans le Guide de l'utilisateur IAM.	Configurer l'accès par programmation en suivant les instructions fournies dans Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur IAM :

- En quoi consiste IAM ?
- Configuration avec IAM
- Création d'un utilisateur et d'un groupe d'administration (console)

Configuration du AWS CLI

Si vous envisagez JupyterLab d'utiliser l' DataBrew API, assurez-vous d'installer le AWS Command Line Interface(AWS CLI). Vous n'en avez pas besoin pour utiliser la DataBrew console ou effectuer les étapes des exercices de mise en route.

Pour configurer le AWS CLI

1. Téléchargez et configurez le AWS CLI en suivant les étapes ci-dessous :
   - Installation de AWS CLI
   - Principes de base de configuration
2. Vérifiez la configuration en saisissant la DataBrew commande suivante à l'invite de commande.

   ```
   aws databrew help
   ```

   Si cette instruction renvoie l'erreur « `aws: error: argument command: Invalid choice` » suivie d'une longue liste de services, désinstallez-les AWS CLI, puis réinstallez-les. Cette action ne remplace pas votre configuration existante.

AWS CLI les commandes utilisent la AWS région par défaut de votre configuration, sauf si vous la définissez avec un paramètre ou un profil. Vous pouvez ajouter le `--region` paramètre à chaque commande.

Si vous préférez, vous pouvez ajouter un profil nommé dans `~/.aws/config` ou `%UserProfile%/.aws/config` (sous Microsoft Windows). Les profils nommés peuvent également conserver d'autres paramètres, comme illustré dans l'exemple suivant.

```
[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
```

Configuration Gestion des identités et des accès AWS Autorisations (IAM)

Avant de commencer, vous devez configurer quelques éléments dans IAM. Vous devez être administrateur ou bénéficier de l'aide d'un administrateur. Toutefois, si vous disposez d'un compte doté d'un accès administrateur, vous pouvez effectuer ces tâches vous-même. Vous trouverez des instructions simples pour chaque tâche dans cette section.

Voici un aperçu de ce que vous devez faire :

- Dans le cadre de ce processus, vous ajoutez un utilisateur. Vous n'êtes pas obligé d'ajouter un nouvel utilisateur, vous pouvez utiliser un utilisateur existant. Vous attachez DataBrew des autorisations afin que l'utilisateur puisse ouvrir la DataBrew console.
- Créez un rôle IAM. Un rôle autorise certaines actions et donne des autorisations lorsqu'il est utilisé, dans certaines limites. Par exemple, cela ne fonctionne que pour les utilisateurs de votre AWS compte. Vous pourrez ajouter d'autres restrictions ultérieurement.
- Créez la ou les politiques IAM dont vous avez besoin. Une politique est une liste de choses qu'un utilisateur est autorisé à faire. Pour créer une politique, vous ouvrez une autre page de console et collez le texte d'un fichier que vous avez téléchargé.

> **Note**
>
> Ce que nous fournissons ici sont des informations de configuration de base. Nous vous recommandons de prendre le temps de personnaliser vos autorisations afin qu'elles répondent à vos besoins en matière de sécurité et de conformité. Si vous avez besoin d'aide, contactez votre administrateur ou le AWS Support.

Pour ajouter les autorisations requises

1. Créez des politiques IAM pour permettre aux utilisateurs de s'exécuter DataBrew en procédant comme suit :

   - Ajoutez une politique IAM personnalisée pour un utilisateur de console. Si vous n'avez pas besoin d'une politique personnalisée, vous pouvez choisir la politique AWS gérée à la place.

Il suffit de l'ajouter à l'utilisateur à l'étape 2. Un utilisateur disposant de ces autorisations peut accéder à la console DataBrew de service.

- Ajoutez des autorisations pour les ressources de données. Un rôle IAM doté de ces autorisations peut accéder aux données au nom de l'utilisateur.

Vous devez être administrateur pour créer des utilisateurs, des rôles et des politiques.

2. Ajoutez des utilisateurs ou des groupes pour DataBrew. Un utilisateur ou un groupe doté des autorisations appropriées peut accéder à la DataBrew console.
3. Ajoutez un rôle avec des autorisations d'accès aux données pour DataBrew. Un rôle doté des autorisations appropriées peut accéder aux données au nom de l'utilisateur.

Configuration de politiques IAM pour DataBrew

Vous utilisez les politiques IAM pour gérer les autorisations. Une politique permet d'ajouter plus facilement les autorisations associées en une seule fois, plutôt qu'une par une.

Nous vous recommandons de créer les politiques en utilisant les mêmes noms que ceux que nous avons fournis. Nous utilisons les noms ci-dessous pour ces politiques dans l'ensemble de la documentation. L'utilisation de ces noms vous facilitera également la tâche si vous devez contacter le AWS Support. Toutefois, vous pouvez choisir de modifier à la fois le nom des politiques et leur contenu. Pour plus d'informations sur les politiques IAM, voir Création d'une politique gérée par le client dans le Guide de l'utilisateur IAM.

Après avoir créé les politiques nécessaires à utiliser DataBrew, vous les associez aux utilisateurs et aux rôles. La procédure à suivre est expliquée plus loin dans cette section.

Rubriques

- Ajouter une politique IAM pour un utilisateur de console
- Ajouter des autorisations pour les ressources de données pour un rôle IAM
- Configuration des politiques IAM pour DataBrew

Ajouter une politique IAM pour un utilisateur de console

La configuration des autorisations d'un utilisateur pour le AWS Management Console est facultative, mais si vous avez besoin d'un accès à la console, procédez d'abord de cette étape.

Pour configurer les autorisations d'accès DataBrew sur la console, choisissez l'une des options suivantes :

- Utilisez la politique gérée par AWS :`AwsGlueDataBrewFullAccessPolicy`. Si vous choisissez cette option, passez à la politique suivante,Ajouter des autorisations pour les ressources de données pour un rôle IAM.
- Créez la politique décrite dans cette section,`AwsGlueDataBrewCustomUserPolicy`. Cette option vous permet de personnaliser la politique avec des exigences de sécurité personnalisées supplémentaires.

La politique suivante accorde les autorisations nécessaires pour exécuter la DataBrew console. Vous fournissez ces autorisations à l'aide d'IAM.

Pour définir la politique AwsGlueDataBrewCustomUserPolicy IAM pour DataBrew (console)

1. Téléchargez le JSON pour la politique `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
3. Dans le panneau de navigation, choisissez Politiques.
4. Pour chaque stratégie, choisissez Create Policy.
5. Sur l'écran Create Policy, accédez à l'onglet JSON.
6. Copiez l'instruction JSON de politique que vous avez téléchargée. Collez-le sur l'exemple de déclaration dans l'éditeur.
7. Vérifiez que la politique est adaptée à votre compte, aux exigences de sécurité et aux AWS ressources requises. Si vous devez apporter des modifications, vous pouvez les faire dans l'éditeur.
8. Choisissez Examiner une politique.

Pour définir la politique AwsGlueDataBrewCustomUserPolicy IAM pour DataBrew (AWS CLI)

1. Téléchargez le JSON pour la politique `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Personnalisez la politique comme décrit dans la première étape de la procédure précédente.
3. Exécutez la commande suivante pour créer la politique.

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

Ajouter des autorisations pour les ressources de données pour un rôle IAM

Pour se connecter aux données, il AWS Glue DataBrew faut disposer d'un rôle IAM qu'il peut transmettre au nom de l'utilisateur. Vous trouverez ci-dessous comment créer la politique que vous associerez ultérieurement à un rôle IAM.

La `AwsGlueDataBrewDataResourcePolicy` politique accorde les autorisations nécessaires pour se connecter aux données à l'aide de DataBrew. Toute opération qui accède aux données d'une autre AWS ressource, telle que l'accès à vos objets dans Amazon S3, DataBrew nécessite une autorisation pour accéder à la ressource en votre nom.

Pour définir la politique AwsGlueDataBrewDataResourcePolicy IAM pour DataBrew (console)

1. Téléchargez le JSON pour `AwsGlueDataBrewDataResourcePolicy`.
2. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
3. Dans le panneau de navigation, choisissez Politiques.
4. Pour chaque stratégie, choisissez Create Policy.
5. Sur l'écran Create Policy, accédez à l'onglet JSON.
6. Copiez l'instruction JSON de politique que vous avez téléchargée. Collez-le sur l'exemple de déclaration dans l'éditeur.
7. Vérifiez que la politique est adaptée à votre compte, aux exigences de sécurité et aux AWS ressources requises. Si vous devez apporter des modifications, vous pouvez les faire dans l'éditeur.
8. Choisissez Examiner une politique.

Pour définir la politique AwsGlueDataBrewDataResourcePolicy IAM pour DataBrew (AWS CLI)

1. Téléchargez le JSON pour `AwsGlueDataBrewDataResourcePolicy`.
2. Personnalisez la politique comme décrit dans la première étape de la procédure précédente.
3. Exécutez la commande suivante pour créer la politique.

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

Configuration des politiques IAM pour DataBrew

Vous trouverez ci-dessous des informations et des exemples sur les politiques IAM que vous pouvez utiliser avec DataBrew. Les détails concernant les politiques de base sont fournis ici. De plus, il existe d'autres exemples dont l'utilisation n'est pas obligatoire DataBrew. Il s'agit de configurations supplémentaires que vous pouvez utiliser dans certaines situations.

Rubriques

AwsGlueDataBrewCustomUserPolicy

La `AwsGlueDataBrewCustomUserPolicy` politique accorde la plupart des autorisations requises pour utiliser la DataBrew console. Certaines des ressources spécifiées dans cette politique font référence à des services utilisés par DataBrew. Il s'agit notamment des AWS Glue Data Catalog noms de compartiments Amazon S3, d'Amazon CloudWatch Logs et de AWS KMS ressources. Elle est similaire à la politique AWS-managed nomméeAwsGlueDataBrewFullAccessPolicy.

Le tableau suivant décrit les autorisations accordées par cette politique.

Action	Ressource	Description
`"databrew:*"`	`"*"`	Accorde l'autorisation d'exécuter toutes les opérations DataBrew d'API.

`"glue:GetDatabases"`
 	
`"glue:GetPartitions"`
 	
`"glue:GetTable"`
	`"*"`	Permet de répertorier les AWS Glue bases de données et les tables.

Action	Ressource	Description
"glue:GetTables" "glue:GetDataCatalogEncryptionSettings"		
"dataexchange:ListDataSets" "dataexchange:ListDataSetRevisions" "dataexchange:ListRevisionAssets" "dataexchange:CreateJob" "dataexchange:StartJob" "dataexchange:GetJob"	"*"	Permet de répertorier les ressources AWS Data Exchange dans les ensembles de données.
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	Permet de répertorier AWS KMS les clés à utiliser pour le chiffrement des résultats de la tâche.
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	Permet de chiffrer le résultat de la tâche.
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionConfiguration"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Permet de répertorier les compartiments Amazon S3 pour les projets, les ensembles de données et les tâches. Permet d'envoyer des fichiers de sortie à S3.

Action	Ressource	Description
`"sts:GetCallerIdentity"`	`"*"`	Obtenez des informations sur l'appelant actuel.
`"cloudtrail:LookupEvents",`	`"*"`	Autoriser la liste AWS CloudTrail des événements pour les ensembles de données (lignage des données).
`"iam:ListRoles"` `"iam:GetRole"`	`"*"`	Permet de répertorier les rôles IAM à utiliser pour les projets et les tâches.

AwsGlueDataBrewDataResourcePolicy

La `AwsGlueDataBrewDataResourcePolicy` politique accorde les autorisations nécessaires pour se connecter aux données et pour effectuer la configuration DataBrew.

Le tableau suivant décrit les autorisations accordées par cette politique.

Action	Ressource	Description
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Vous permet de prévisualiser vos fichiers.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permet d'envoyer des fichiers de sortie à S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Permet de supprimer un objet créé par DataBrew.

Action	Ressource	Description
"s3:ListBucket"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Permet de répertorier les compartiments Amazon S3 issus de projets, d'ensembles de données et de tâches.
"kms:Decrypt"	"arn:aws:kms:::key/key_ids"	Permet de déchiffrer les ensembles de données chiffrés.
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	Permet de chiffrer le résultat de la tâche.
"ec2:DescribeVpcEndpoints" "ec2:DescribeRouteTables" "ec2:DeleteNetworkInterface" "ec2:DescribeNetworkInterfaces" "ec2:DescribeSecurityGroups" "ec2:DescribeSubnets" "ec2:DescribeVpcAttribute" "ec2:CreateNetworkInterface"	"*"	Permet de configurer des éléments du réseau Amazon EC2, tels que des clouds privés virtuels (VPC), lors de l'exécution de tâches et de projets.

Action	Ressource	Description
`"ec2:DeleteNetwork Interface"`	`"*"`	Permet de supprimer une interface réseau dans un VPC.

`"ec2:CreateTags"`
 	
`"ec2:DeleteTags"`
	`"arn:aws:ec2:::net work-interface/*", "arn:aws:ec2:::sec urity-group/*"`	Permet de créer et de supprimer des balises.
		
		Vous avez besoin de ces autorisations si vous utilisez un catalogue de AWS Glue données avec un VPC activé. DataBrew transmet des données AWS Glue pour exécuter vos tâches et vos projets. Ces autorisat ions permettent de baliser les ressources Amazon EC2 créées pour les points de terminaison de développe ment.AWS Glueétiquette les interfaces réseau, les groupes de sécurité et les instances Amazon EC2 avec. `aws-glue-service-resource`

`"logs:CreateLogGroup"`
 	
`"logs:CreateLogStream"`
 	
`"logs:PutLogEvents"`
	`"arn:aws:logs:::lo g-group:/aws-glue-databrew/*"`	Permet d'écrire des journaux sur Amazon CloudWatch Logs
		
		DataBrew écrit des journaux dans des groupes de journaux dont le nom commence par`aws-glue-databrew` .

Action	Ressource	Description
`"lakeformation:GetDataAccess"`	`"*"`	Permet l'accès àAWS Lake Formation, à condition `"Glue":"GetTable"` que ce soit également autorisé L'utilisation de Lake Formation nécessite une configuration supplémentaire dans la console Lake Formation.

Politique IAM pour utiliser les objets Amazon S3 avec DataBrew

La `AwsGlueDataBrewSpecificS3BucketPolicy` politique accorde les autorisations nécessaires pour accéder à S3 au nom des utilisateurs non administratifs.

Personnalisez la politique comme suit :

1. Remplacez les chemins Amazon S3 dans la politique afin qu'ils pointent vers les chemins que vous souhaitez utiliser. Dans l'exemple de texte, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* représente un objet ou un fichier spécifique. *`BUCKET-NAME-2/`*représente tous les objets (*) dont le nom de chemin commence par`BUCKET-NAME-2/`. Mettez-les à jour pour nommer les buckets que vous utilisez.
2. (Facultatif) Utilisez des caractères génériques dans les chemins Amazon S3 pour restreindre davantage les autorisations. Pour plus d'informations, consultez Éléments des politiques IAM : variables et balises dans le Guide de l'utilisateur IAM.

   Bonnes pratiques en matière de sécurité : pour empêcher tout accès non autorisé aux compartiments Amazon S3 portant des noms similaires dans d'autres AWS comptes, incluez la clé de `aws:ResourceAccount` condition dans votre politique. Cela garantit que DataBrew vous ne pouvez accéder aux compartiments que dans votre propre AWS compte, même si vous utilisez des ARN de ressources génériques. Ajoutez la condition suivante à vos déclarations de politique :

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

```
}
}
```

`123456789012`Remplacez-le par votre numéro de AWS compte actuel.

Dans ce cadre, vous pouvez restreindre les autorisations pour les actions `s3:PutObject` et`s3:PutBucketCORS`. Ces actions ne sont requises que pour les utilisateurs qui créent DataBrew des projets, car ces utilisateurs doivent être en mesure d'envoyer des fichiers de sortie à S3.

Pour plus d'informations et pour découvrir des exemples de ce que vous pouvez ajouter à une politique IAM pour Amazon S3, consultez la section Exemples de politiques relatives aux compartiments dans le guide du développeur Amazon S3.

Le tableau suivant décrit les autorisations accordées par cette politique.

Action	Ressource	Description
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Vous permet de prévisualiser vos fichiers.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permet d'envoyer des fichiers de sortie à S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permet de supprimer un objet.

Pour définir la politique AwsGlueDataBrewSpecificS3BucketPolicy IAM pour DataBrew (console)

1. Téléchargez le JSON pour la politique `AwsGlueDataBrewSpecificS3BucketPolicy`IAM.

2. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
3. Dans le panneau de navigation, choisissez Politiques.
4. Pour chaque stratégie, choisissez Create Policy.
5. Sur l'écran Create Policy, accédez à l'onglet JSON.
6. Collez l'instruction JSON de politique sur l'exemple d'instruction dans l'éditeur.
7. Vérifiez que la politique est adaptée à votre compte, aux exigences de sécurité et aux AWS ressources requises. Si vous devez apporter des modifications, vous pouvez les faire dans l'éditeur.
8. Choisissez Examiner une politique.

Pour définir la politique AwsGlueDataBrewSpecificS3BucketPolicy IAM pour DataBrew (AWS CLI)

1. Téléchargez le JSON pour `AwsGlueDataBrewSpecificS3BucketPolicy`.
2. Personnalisez la politique comme décrit dans la première étape de la procédure précédente.
3. Exécutez la commande suivante pour créer la politique.

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

Politique IAM pour utiliser le chiffrement avec DataBrew

La `AwsGlueDataBrewS3EncryptedPolicy` politique accorde les autorisations nécessaires pour accéder aux objets S3 chiffrés avec AWS Key Management Service(AWS KMS) au nom des utilisateurs non administrateurs.

Personnalisez la politique comme suit :

1. Remplacez les chemins Amazon S3 dans la politique afin qu'ils pointent vers les chemins que vous souhaitez utiliser. Dans l'exemple de texte, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* représente un objet ou un fichier spécifique. *`BUCKET-NAME-2/`*représente tous les objets (*) dont le nom de chemin commence par`BUCKET-NAME-2/`. Mettez-les à jour pour nommer les buckets que vous utilisez.

2. (Facultatif) Utilisez des caractères génériques dans les chemins Amazon S3 pour restreindre davantage les autorisations. Pour en savoir plus, consultez Éléments des politiques IAM : variables et balises.

   Dans ce cadre, vous pouvez restreindre les autorisations pour les actions `s3:PutObject` et`s3:PutBucketCORS`. Ces actions ne sont requises que pour les utilisateurs qui créent DataBrew des projets, car ces utilisateurs doivent être en mesure d'envoyer des fichiers de sortie à S3.

   Pour plus d'informations et pour voir des exemples de ce que vous pouvez ajouter à une politique IAM pour Amazon S3, consultez Exemples de politiques de compartiment.

3. Recherchez les ARN de ressources suivants dans le `ToUseKms` fichier.

```
  "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. Remplacez le AWS compte d'exemple par votre numéro de AWS compte (sans tirets).
5. Modifiez la liste d'exemples pour répertorier à la place les rôles IAM que vous souhaitez utiliser. Nous vous recommandons de définir la portée de vos politiques IAM en fonction du plus petit ensemble d'autorisations possible. Toutefois, vous pouvez autoriser votre utilisateur à accéder à tous les rôles IAM, par exemple si vous utilisez un compte d'apprentissage personnel avec des exemples de données. Pour permettre à la liste d'accéder à tous les rôles IAM, remplacez la liste d'exemples par une seule entrée :`"arn:aws:iam::111122223333:role/*"`.

Le tableau suivant décrit les autorisations accordées par cette politique.

Action	Ressource	Description
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Vous permet de prévisualiser vos fichiers.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permet de répertorier les compartiments Amazon S3 issus de projets, d'ensembles de données et de tâches.

Action	Ressource	Description
`"s3:PutObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permet d'envoyer des fichiers de sortie à S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Permet de supprimer un objet créé par DataBrew.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Permet de déchiffrer les ensembles de données chiffrés.
`"kms:GenerateDataKey*"`	`"arn:aws:kms:::key/key_ids"`	Permet de chiffrer le résultat de la tâche.

Pour définir la politique AwsGlueDataBrewS3EncryptedPolicy IAM pour DataBrew (console)

1. Téléchargez le JSON pour la politique `AwsGlueDataBrewS3EncryptedPolicy`IAM.
2. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/l'adresse.
3. Dans le panneau de navigation, choisissez Politiques.
4. Pour chaque stratégie, choisissez Create Policy.
5. Sur l'écran Create Policy, accédez à l'onglet JSON.
6. Collez l'instruction JSON de politique sur l'exemple d'instruction dans l'éditeur.
7. Vérifiez que la politique est adaptée à votre compte, aux exigences de sécurité et aux AWS ressources requises. Si vous devez apporter des modifications, vous pouvez les faire dans l'éditeur.
8. Choisissez Examiner une politique.

Pour définir la politique AwsGlueDataBrewS3EncryptedPolicy IAM pour DataBrew (AWS CLI)

1. Téléchargez le JSON pour `AwsGlueDataBrewS3EncryptedPolicy`.
2. Personnalisez la politique comme décrit dans la première étape de la procédure précédente.
3. Exécutez la commande suivante pour créer la politique.

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

Ajouter des utilisateurs ou des groupes dotés d' DataBrew autorisations

Vous attribuez des politiques aux rôles, et des rôles aux utilisateurs et aux groupes pour gérer les autorisations. Pour plus d'informations, consultez la section Identités IAM (utilisateurs, groupes et rôles) dans le guide de l'utilisateur IAM.

Avant de commencer, vous devez avoir au moins un utilisateur auquel attribuer des autorisations.

Utilisez la procédure suivante pour configurer DataBrew les autorisations pour les utilisateurs qui doivent travailler dans la DataBrew console ou exécuter des DataBrew commandes dans la CLI.

Pour configurer les DataBrew autorisations

1. Créez une clé d'accès permettant à votre utilisateur d'utiliser le AWS CLI for DataBrew et d'autres outils de développement.
2. Activez AWS Management Console l'accès pour permettre à l'utilisateur d'utiliser la AWS console.
3. Créez un rôle pour les DataBrew utilisateurs ou les groupes.
4. Choisissez la politique que vous utilisez. Effectuez l’une des actions suivantes :
   - Si vous l'avez créé`AwsGlueDataBrewCustomUserPolicy`, sélectionnez-le dans la liste.
   - Pour utiliser la AWS-managed politique, sélectionnez-la `AwsGlueDataBrewFullAccessPolicy` dans la liste.
5. Attribuez cette politique au rôle.
6. Définissez les relations de confiance pour le rôle afin qu'un utilisateur ou un groupe puisse assumer le rôle approprié.
   - Si vous n'utilisez pas de groupes, confiez le rôle à l'utilisateur.

- Si vous utilisez des groupes, confiez le rôle au groupe et ajoutez-y l'utilisateur.

Ajouter un rôle IAM avec des autorisations de ressources de données

Vous utilisez les rôles IAM pour gérer les politiques attribuées ensemble. Un rôle IAM peut être utilisé par une personne agissant dans un rôle particulier, par exemple un DataBrew utilisateur ou DataBrew lui-même. Pour plus d'informations, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM.

Utilisez la procédure suivante pour créer un rôle IAM requis pour que les DataBrew projets puissent accéder aux données.

Pour associer la politique IAM requise à un nouveau rôle IAM pour DataBrew

1. Dans le volet de navigation, choisissez Rôles, puis Créer un rôle.
2. Pour Sélectionner le type d'entité de confiance, choisissez le AWS service étiqueté par carte.
3. DataBrewChoisissez dans la liste, puis cliquez sur Suivant : Autorisations.
4. Entrez **`AwsGlueDataBrewDataResourcePolicy`** dans le champ de recherche (la politique IAM que vous avez créée lors d'une étape précédente). Sélectionnez la politique et choisissez Next : Tags.
5. Choisissez Suivant : Vérification.
6. Pour Nom du rôle, saisissez **`AwsGlueDataBrewDataAccessRole`**, puis choisissez Créer un rôle.

Configuration AWS IAM Identity Center(Centre d'identité IAM)

À l'aide de AWS IAM Identity Center(IAM Identity Center), vos utilisateurs peuvent se connecter à l' DataBrew aide d'une simple URL, sans se connecter AWS Management Console et sans avoir besoin d'un AWS compte.

Pour configurer IAM Identity Center

1. Ouvrez la AWS Organizations console et créez une organisation si vous n'en avez pas déjà une. Toutes les fonctionnalités sont activées par défaut pour cette organisation.

   Pour plus d'informations, consultez AWS IAM Identity Center les sections Conditions préalables et Création et gestion d'une organisation.

2. Ouvrez la console AWS IAM Identity Center.
3. Choisissez votre source d'identité.

   Par défaut, vous disposez d'un magasin IAM Identity Center pour une gestion rapide et facile des utilisateurs. Vous pouvez éventuellement connecter un fournisseur d'identité externe à la place ou connecter un AWS Managed Microsoft AD annuaire à votre Active Directory local. Dans ce guide, nous utilisons le magasin IAM Identity Center par défaut.

   Pour plus d'informations, voir Choisir votre source d'identité dans le Guide de AWS IAM Identity Center l'utilisateur.
4. Créez un ensemble d'autorisations d' DataBrew accès :

   a. Dans le volet de navigation d'IAM Identity Center, choisissez AWS des comptes, puis choisissez Ensembles d'autorisations.

   b. Sur la page Créer un ensemble d'autorisations, choisissez Créer un ensemble d'autorisations personnalisé.

   c. Pour État du relais, entrez`https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`.

      Cette saisie permet à vos utilisateurs d'accéder directement à DataBrew.

   d. Choisissez Joindre les politiques AWS gérées DataBrew, recherchez et choisissez AwsGlueDataBrewFullAccessPolicy. Cette option donne à vos utilisateurs toutes les autorisations dont ils ont besoin DataBrew. Vous trouverez plus de détails dansAjouter une politique IAM pour un utilisateur de console.

   e. (Facultatif) Choisissez Créer une politique d'autorisation personnalisée et personnalisez les autorisations pour vos utilisateurs.
5. Dans le volet de navigation d'IAM Identity Center, choisissez Groups, puis Create group. Entrez le nom du groupe et choisissez Create.
6. Ajoutez un utilisateur au magasin IAM Identity Center :

   a. Dans le volet de navigation d'IAM Identity Center, sélectionnez Users.

   b. Sur l'écran Ajouter un utilisateur, entrez les informations requises et choisissez Envoyer un e-mail à l'utilisateur avec les instructions de configuration du mot de passe. L'utilisateur doit recevoir un e-mail concernant les prochaines étapes de configuration.

   c. Choisissez Suivant : Groupes, choisissez le groupe de votre choix, puis choisissez Ajouter un utilisateur.

Les utilisateurs devraient recevoir un e-mail les invitant à utiliser le SSO. Dans cet e-mail, ils doivent choisir Accepter l'invitation et définir le mot de passe. Ils peuvent également trouver l'URL du portail dans l'e-mail. Ils peuvent utiliser cette URL pour y accéder DataBrew.

7. Attribuez un compte à chaque utilisateur :

   a. Ouvrez la console IAM Identity Center, puis dans le volet de navigation, sélectionnez les AWS comptes.

   b. Choisissez une AWS organisation et choisissez un AWS compte.

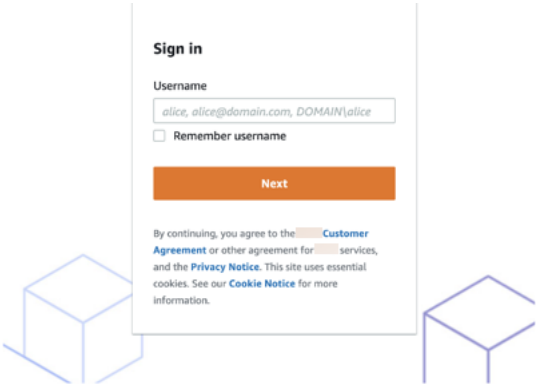
   c. Sur l'écran Attribuer des utilisateurs, choisissez l'onglet Groupes et choisissez le groupe de votre choix.

   d. Choisissez Next: Permission sets (Suivant : Jeux d'autorisations).

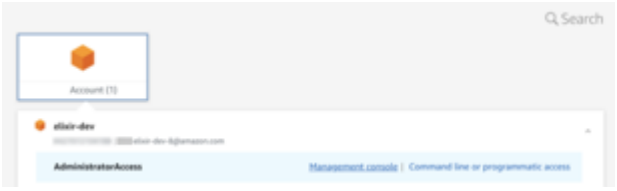
   e. Choisissez l'autorisation définie pour DataBrew, puis cliquez sur Terminer.

Étapes de connexion pour un utilisateur d'IAM Identity Center-enabled

1. Connectez-vous àAWS l'aide d'un Center-enabled compte IAM Identity.



2. Cliquez sur Identité AWS du compte



3. Cliquez sur Console de gestion pour être redirigé vers la console en un clic. DataBrew

Utilisation en DataBrew tant qu'extension dans JupyterLab

> **Warning**
>
> AWS Glue DataBrew JupyterLab le support d'extension prend fin le 31 décembre 2024, car JupyterLab 3 atteindront la fin du support. Pour plus d'informations, voir JupyterLab 3 fin de maintenance.

Si vous préférez préparer les données dans un environnement Jupyter Notebook, vous pouvez utiliser toutes les fonctionnalités d'AWS Glue DataBrew in. JupyterLab

JupyterLab est un environnement de développement interactif basé sur le Web pour Jupyter Notebook. Sur la JupyterLab page Web locale, vous pouvez ajouter des sections pour un terminal, une session SQL, Python, etc. Après avoir installé l'AWS Glue DataBrew extension, vous pouvez ajouter une section pour la DataBrew console. Il fonctionne avec tous les blocs-notes existants ou les autres extensions que vous possédez déjà, directement depuis l' JupyterLab environnement.

Rubriques

Conditions préalables

Avant de commencer, configurez les éléments suivants :

- Un AWS compte — Si vous n'en avez pas encore, commencez parConfiguration d'un nouveau AWS compte.
- Un utilisateur Gestion des identités et des accès AWS(IAM) ayant accès aux autorisations nécessaires pour DataBrew — Pour plus d'informations, voirAjouter des utilisateurs ou des groupes dotés d' DataBrew autorisations.
- Un rôle IAM à utiliser dans les DataBrew opérations : vous pouvez utiliser le rôle par défaut, s'il `AwsGlueDataBrewDataAccessRole` est configuré. Pour configurer des rôles IAM supplémentaires, consultezAjouter un rôle IAM avec des autorisations de ressources de données.

- Une JupyterLab installation (version 2.2.6 ou supérieure) — Pour plus d'informations, consultez les rubriques suivantes de la JupyterLabdocumentation :
  - JupyterLab prérequis
  - JupyterLab installation — Nous vous recommandons d'utiliser`pip install jupyterlab`.
- Une Node.js installation (version 12.0 ou supérieure).
- Une installation AWS Command Line Interface(AWS CLI) — Pour plus d'informations, consultezConfiguration du AWS CLI.
- Installation d'un proxy AWS Jupyter (`pip install aws-jupyter-proxy`) — Cette extension est utilisée avec un point de terminaison de AWS service pour transmettre vos AWS informations d'identification en toute sécurité. Pour plus d'informations, consultez aws-jupyter-proxy on. GitHub

Pour vérifier que les prérequis sont installés, vous pouvez exécuter un test similaire au suivant sur la ligne de commande, comme illustré dans l'exemple suivant.

```
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
```

Le résultat devrait ressembler à ce qui suit. Les répertoires varient en fonction du système d'exploitation et de la configuration.

```
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
```

Configuration JupyterLab pour utiliser l'extension

Après l'installation JupyterLab, vous devez le configurer pour sécuriser l'accès aux données et activer les extensions de serveur.

Pour configurer un mot de passe et un chiffrement

1. Définissez un mot de passe pour protéger les données que vous souhaitez ajouter dans l'extension. Jupyter fournit un mot de passe utilitaire. Exécutez la commande suivante et entrez le mot de passe de votre choix à l'invite.

   ```
   jupyter notebook password
   ```

   Le résultat se présente comme suit.

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

2. Activez le chiffrement sur le serveur Jupyter. Si vous installez Jupyter sur votre machine locale et que personne ne peut y accéder via le réseau, vous pouvez ignorer cette étape.

   Pour configurer le chiffrement avec le protocole TLS (Transport Layer Security), créez un certificat personnalisé pour votre environnement. Pour plus d'informations, voir Utilisation de Let's Encrypt pour sécuriser un serveur dans la documentation Jupyter.

3. Pour commencer JupyterLab, exécutez la commande suivante à l'invite de commande.

```
jupyter lab
```

Pour plus d'informations, consultez la section Démarrage JupyterLab dans la JupyterLab documentation.

4. Pendant JupyterLab l'exécution, vous pouvez y accéder via une URL similaire à la suivante : http://localhost:8888/lab. Si vous configurez le chiffrement, utilisez-le à la `https` place de`http`. Si vous avez personnalisé le port, remplacez-le par votre numéro de port par`8888`.

Suivez la procédure ci-dessous pour activer les extensions tierces.

Pour activer les extensions tierces dans JupyterLab

1. Sur la JupyterLab page Web, choisissez l'icône du gestionnaire d'extensions dans le menu de gauche.
2. Lisez l'avertissement concernant les risques liés à l'exécution d'extensions tierces. N'installez que des extensions provenant de développeurs en qui vous avez confiance.
3. Pour activer les extensions tierces dans JupyterLab, choisissez Activer.
4. Suivez les instructions pour reconstruire et recharger JupyterLab.

Activation de l' DataBrew extension pour JupyterLab

Après avoir effectué une installation sécurisée JupyterLab avec les extensions activées, installez l' DataBrew extension afin de pouvoir l'exécuter DataBrew dans votre bloc-notes.

Pour installer les extensions pour DataBrew (console)

1. Pour commencer JupyterLab, exécutez la commande suivante à l'invite de commande.

```
jupyter lab
```

2. Sur la JupyterLab page Web, choisissez l'icône du gestionnaire d'extensions dans le menu de gauche.
3. Recherchez l' DataBrew extension en saisissant « **`brew`** » pour Rechercher en haut à gauche.
4. Localisez aws_glue_databrew_jupyter dans la liste, mais ne cliquez pas dessus. Si vous cliquez sur le nom surligné de l'extension, une nouvelle fenêtre de navigateur s'ouvre avec la page aws_glue_databrew_jupyter activée. GitHub

5. Pour installer l' DataBrew extension, choisissez l'une des options suivantes :

   - Sur la ligne de commande, exécutez`jupyter labextension install aws_glue_databrew_jupyter`.
   - Choisissez Installer en bas de la carte d'extension, sous « aws_glue_databrew_jupyter » en lettres grises.

   DataBrew l'extension est compatible avec les JupyterLab versions 1.2 et 2.x.
6. Pour vérifier qu'il est installé, exécutez`jupyter labextension list`. Le résultat devrait ressembler à ce qui suit.

   ```
   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. Reconstruisez JupyterLab en utilisant l'une des méthodes suivantes :

   - À l'invite de commande, exécutez`jupyter lab build`.
   - Sur la page Web, choisissez Reconstruire en haut à gauche.
8. Lorsque le build est terminé, effectuez l'une des opérations suivantes :

   - À l'invite de commande, exécutez`jupyter lab`.
   - Sur la page Web, choisissez Recharger dans le message Build Complete.
9. Sur la JupyterLab page Web, fermez le gestionnaire d'extensions en choisissant son icône dans le menu de gauche.

   Pour ouvrir l'extension, choisissez Lancer dans la section Autre AWS Glue DataBrew de l'onglet Lanceur. L'extension utilise votre AWS CLI configuration actuelle pour les clés d'accès et les paramètres AWS régionaux.

Une fois la configuration terminée, vous pouvez utiliser l'AWS Glue DataBrewonglet pour interagir DataBrew de l'intérieur JupyterLab.

Démarrage avec AWS Glue DataBrew

Vous pouvez utiliser le didacticiel suivant pour vous guider dans la création de votre premier DataBrew projet. Vous chargez un exemple de jeu de données, vous exécutez des transformations sur cet ensemble de données, vous créez une recette pour capturer ces transformations et vous exécutez une tâche pour écrire les données transformées sur Amazon S3.

Rubriques

Conditions préalables

Avant de continuer, suivez les instructions applicables dansConfiguration AWS Glue DataBrew. Continuez ensuite versÉtape 1 : Créer un projet.

Étape 1 : Créer un projet

Au cours de cette étape, vous pouvez utiliser la DataBrew console pour démarrer rapidement avec un exemple de projet.

Pour créer un projet

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.
2. Assurez-vous que votre AWS région est sélectionnée dans le coin supérieur droit de la DataBrew console. Pour obtenir la liste des AWS régions prises en charge par DataBrew, voir les DataBrew points de terminaison et les quotas dans le Références générales AWS.
3. Dans le volet de navigation, choisissez Projects, puis Create project.

4. Dans le volet Détails du projet, procédez comme suit :

   - Dans le champ Nom du projet, entrez`chess-project`.
   - Pour la recette ci-jointe, créez une nouvelle recette. Un nom suggéré pour la recette est fourni (`chess-project-recipe`).
5. Dans le volet Sélectionner un jeu de données, choisissez Sample files.
6. Dans le volet Exemples de fichiers, choisissez Famous chess game moves. Ce jeu de données contient des informations détaillées sur plus de 20 000 parties d'échecs.

   Pour le nom du jeu de données, un nom suggéré pour l'ensemble de données est fourni (`chess-games`).
7. Dans le volet Autorisations d'accès, choisissez`AwsGlueDataBrewDataAccessRole`. Il s'agit d'un rôle lié à un service qui permet DataBrew d'accéder à vos compartiments Amazon S3 en votre nom.
8. Choisissez Créer un projet et attendez la DataBrew fin de la préparation du projet. La fenêtre ressemble à la fenêtre suivante.

   Les données que vous voyez représentent un échantillon de l'`chess-games`ensemble de données. Par défaut, l'échantillon comprend les 500 premières lignes de l'ensemble de données. Vous pourrez modifier ce paramètre de projet ultérieurement.

   La barre d'outils donne accès à des centaines de transformations de données que vous pouvez appliquer aux données.

   Le volet des recettes situé à droite de la DataBrew console suit les transformations que vous avez appliquées jusqu'à présent.

Étape 2 : Résumez les données

Au cours de cette étape, vous allez créer une DataBrew recette, c'est-à-dire un ensemble de transformations qui peuvent être appliquées à cet ensemble de données et à d'autres jeux similaires. Lorsque la recette est terminée, vous la publiez afin qu'elle puisse être utilisée.

Au jeu d'échecs, les joueurs peuvent être évalués en fonction de leurs performances face aux autres joueurs. (Pour plus d'informations, consultez https://en.wikipedia.org/wiki/Chess_rating_system). Dans ce didacticiel, vous vous concentrez uniquement sur les jeux où les deux joueurs étaient de classe A, ce qui signifie que leur note était de 1800 ou plus.

Pour résumer les données

1. Dans la barre d'outils de transformation, choisissez Filtrer, Par condition, Supérieur ou égal à.
2. Définissez ces options comme suit :

   - Colonne source - `white_rating`
   - État du filtre : supérieur ou égal à 1800

   Pour voir comment fonctionne la transformation, choisissez Aperçu des modifications. Choisissez ensuite Appliquer.
3. Répétez l'étape précédente, mais cette fois, définissez la colonne Source sur`black_rating`. Une fois que vous avez appliqué vos modifications, les exemples de données contiennent uniquement les parties où les joueurs de chaque côté (noir et blanc) appartenaient à la classe A ou à un niveau supérieur.
4. Résumez les données pour déterminer combien de parties ont été gagnées par chaque camp. Pour ce faire, dans la barre d'outils de transformation, choisissez Grouper.
5. Pour les propriétés du groupe, procédez comme suit :

   a. Dans la première ligne, choisissez `winner` le nom de colonne. Laissez Aggregate défini sur Grouper par.

   b. Dans la deuxième ligne, choisissez `victory_status` le nom de colonne. Laissez Aggregate défini sur Grouper par.

   c. Choisissez Ajouter une autre colonne.

   d. Dans la troisième ligne, choisissez `winner` le nom de colonne. Réglez Aggregate sur Count.

   e. Pour Type de groupe, choisissez Grouper comme nouvelle table. Le volet d'aperçu vous montre à quoi ressemblera le résultat.

   f. Choisissez Finish (Terminer).
6. Choisissez Publier pour enregistrer votre travail, à droite dans le volet des recettes.
7. Dans le champ Description de la version, saisissez Première version de ma recette. Choisissez ensuite Publier.

Étape 3 : ajouter d'autres transformations

Au cours de cette étape, vous ajoutez d'autres transformations à votre recette et publiez une autre version de celle-ci. Pour affiner notre exemple, nous utilisons l'information selon laquelle toutes les parties d'échecs ne se terminent pas par un gagnant clair ; certaines parties sont jouées jusqu'à un match nul.

Pour ajouter d'autres transformations de recettes et les republier

1. Dans la barre d'outils de transformation, choisissez Filtrer, par condition, est pour ne pas supprimer les parties jouées pour un match nul.
2. Définissez ces options comme suit :

   - Colonne source - `victory_status`
   - État du filtre — Non `draw`

   Pour ajouter cette transformation à votre recette, choisissez Appliquer.
3. Modifiez les données `victory_status` pour qu'elles soient plus pertinentes. Pour ce faire, dans la barre d'outils de transformation, choisissez Nettoyer, Remplacer, Remplacer la valeur ou le modèle.
4. Définissez ces options comme suit :

   - Colonne source - `victory_status`
   - Spécifier les valeurs à remplacer : valeur ou modèle
   - Valeur à remplacer - `mate`
   - Remplacer par la valeur - `checkmate`

   Pour ajouter cette transformation à votre recette, choisissez Appliquer.
5. Répétez l'étape précédente, mais passez `resign` à`other player resigned`.
6. Répétez l'étape précédente, mais passez `outoftime` à`time ran out`.
7. Choisissez Publier pour enregistrer votre travail, à droite dans le volet des recettes.

Étape 4 : Passez en revue vos DataBrew ressources

Maintenant que vous avez travaillé sur un exemple de projet, passez en revue les DataBrew ressources que vous avez créées jusqu'à présent.

Pour passer en revue vos DataBrew ressources

1. Dans le volet de navigation, sélectionnez Datasets.

   Lorsque vous avez créé l'exemple de projet, vous avez DataBrew créé un ensemble de données pour vous (`chess-games`). Le fichier de données source est stocké dans Amazon S3 et est au format Microsoft Excel (`chess-games.xlsx`). Le fichier contient les métadonnées de plus de 20 000 parties d'échecs. L'`chess-games`ensemble de données fournit les informations DataBrew nécessaires pour lire les données de ce fichier.
2. Dans le volet de navigation, sélectionnez Projects.

   Vous devriez voir le projet sur lequel vous avez travaillé dans les étapes précédentes (`chess-project`). Chaque projet nécessite un jeu de données, dans ce cas`chess-games`. Chaque projet nécessite également une recette, afin que vous puissiez ajouter des étapes de transformation des données au fur et à mesure. Lorsque vous avez créé cet exemple de projet, vous avez DataBrew créé une nouvelle recette (vide) pour vous et l'avez jointe au projet.
3. Dans le volet de navigation, choisissez Recipes, puis dans la colonne Nom de la recette, choisissez chess-project-recipe. Cela vous montre la recette DataBrew créée pour votre projet et que vous avez affinée en y ajoutant des étapes de transformation.
4. À gauche, consultez les versions de recettes qui ont été publiées. Choisissez l'une d'entre elles pour afficher son onglet Étapes de recette, qui affiche les détails de la recette et les étapes pour cette version.
5. Consultez l'onglet Data Lineage, qui indique d'où proviennent les données et comment elles sont utilisées. Pour plus de détails, choisissez l'une des icônes du diagramme.

Étape 5 : Création d'un profil de données

Lorsque vous travaillez sur un projet, DataBrew affiche des statistiques telles que le nombre de lignes de l'échantillon et la distribution des valeurs uniques dans chaque colonne. Ces statistiques, et bien d'autres encore, représentent un profil de l'échantillon.

Pour demander un profil de données, créez et exécutez une tâche de profilage.

Pour profiler un ensemble de données

1. Dans le volet de navigation, sélectionnez Jobs.
2. Dans l'onglet Profile jobs, sélectionnez Create job.
3. Dans Nom du Job, entrez`chess-data-profile`.
4. Pour Type de tâche, choisissez Créer une tâche profilée.
5. Dans le volet de saisie Job, procédez comme suit :
   - Pour Exécuter, choisissez Dataset.
   - Choisissez Sélectionner un jeu de données pour afficher la liste des jeux de données disponibles, puis choisissez`chess-games`.
6. Dans le volet des paramètres de sortie du Job, procédez comme suit :
   - Pour Type de fichier, choisissez JSON (JavaScript Object Notation).
   - Choisissez l'emplacement S3 pour afficher la liste des compartiments Amazon S3 disponibles, puis choisissez le compartiment à utiliser. Choisissez ensuite Parcourir. Dans la liste des dossiers, choisissez`databrew-output`, puis sélectionnez Sélectionner.
7. Dans le volet Autorisations d'accès, choisissez`AwsGlueDataBrewDataAccessRole`. Il s'agit d'un rôle lié à un service qui permet DataBrew d'accéder à vos compartiments Amazon S3 en votre nom.
8. Choisissez Create and run job. DataBrew crée une tâche avec vos paramètres, puis l'exécute.
9. Dans le volet Historique d'exécution des tâches, attendez que le statut de la tâche passe de `Running` à`Succeeded`.
10. Pour consulter le profil, choisissez AFFICHER LE PROFIL :

    La fenêtre DATASETS s'affiche. Prenez le temps d'explorer les onglets suivants :
    - Aperçu du jeu de données
    - Vue d'ensemble du profil
    - Statistiques de colonne
    - Statistiques de lignage des données

Étape 6 : Transformer le jeu de données

Jusqu'à présent, vous n'avez testé votre recette que sur un échantillon de l'ensemble de données. Il est maintenant temps de transformer l'ensemble de données en créant une tâche de DataBrew recette.

Lorsque la tâche s'exécute, DataBrew applique votre recette à toutes les données de l'ensemble de données et écrit les données transformées dans un compartiment Amazon S3. Les données transformées sont distinctes du jeu de données d'origine. DataBrew ne modifie pas les données source.

Avant de continuer, assurez-vous que votre compte comporte un compartiment Amazon S3 dans lequel vous pouvez écrire. Dans ce compartiment, créez un dossier pour capturer le résultat de la tâche DataBrew. Pour effectuer ces étapes, procédez comme suit.

Pour créer un compartiment et un dossier S3 afin de capturer le résultat d'une tâche

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.

   Si un compartiment Amazon S3 est déjà disponible et que vous disposez d'autorisations d'écriture pour celui-ci, sautez l'étape suivante.
2. Si vous n'avez pas de compartiment Amazon S3, choisissez Create bucket. Dans le champ Nom du compartiment, entrez un nom unique pour votre nouveau compartiment. Choisissez Créer un compartiment.
3. Dans la liste des compartiments, choisissez celui que vous souhaitez utiliser.
4. Choisissez Créer un dossier.
5. Dans Nom du dossier`databrew-output`, entrez et choisissez Créer un dossier.

Après avoir créé un compartiment et un dossier Amazon S3 pour contenir la tâche, exécutez-la en suivant la procédure suivante.

Pour créer et exécuter une tâche de recette

1. Dans le volet de navigation, sélectionnez Jobs.
2. Dans l'onglet Recipe jobs, choisissez Create job.
3. Dans Nom du Job, entrez`chess-winner-summary`.

4. Dans Type de tâche, choisissez Créer une tâche de recette.
5. Dans le volet de saisie Job, procédez comme suit :

   - Pour Exécuter, choisissez Dataset.
   - Choisissez Sélectionner un jeu de données pour afficher la liste des jeux de données disponibles, puis choisissez`chess-games`.
   - Choisissez Sélectionnez une recette pour afficher la liste des recettes disponibles, puis choisissez`chess-project-recipe`.
6. Dans le volet des paramètres de sortie du Job, procédez comme suit :

   - Type de fichier : choisissez CSV (valeurs séparées par des virgules).
   - Emplacement S3 : choisissez ce champ pour afficher la liste des compartiments Amazon S3 disponibles, puis choisissez le compartiment à utiliser. Choisissez ensuite Parcourir. Dans la liste des dossiers, sélectionnez`databrew-output`, puis sélectionnez Sélectionner.
7. Dans le volet Autorisations d'accès, choisissez`AwsGlueDataBrewDataAccessRole`. Ce rôle lié à un service permet DataBrew d'accéder à vos compartiments Amazon S3 en votre nom.
8. Choisissez Create and run job. DataBrew crée une tâche avec vos paramètres, puis l'exécute.
9. Dans le volet Historique d'exécution des tâches, attendez que le statut de la tâche passe de `Running` à`Succeeded`.
10. Choisissez Output pour accéder à la console Amazon S3. Choisissez votre compartiment S3, puis le `databrew-output` dossier pour accéder à la sortie de la tâche.
11. (Facultatif) Choisissez Télécharger pour télécharger le fichier et afficher son contenu.

Étape 7 : (Facultatif) Nettoyer

La procédure pas à pas est terminée. Vous pouvez continuer à utiliser DataBrew les ressources Amazon S3 que vous avez créées ou les supprimer.

Pour nettoyer des ressources

1. Ouvrez la DataBrew console à https://console.aws.amazon.com/databrew/, puis dans le volet de navigation, sélectionnez Projects.
2. Choisissez votre projet (exemple de projet). Pour Actions, choisissez Supprimer.
3. Dans le volet Supprimer un exemple de projet, choisissez Supprimer la recette jointe. Ensuite, choisissez Supprimer. Votre projet, ainsi que sa recette et ses tâches, seront supprimés.

4. Dans le volet de navigation, sélectionnez Datasets.
5. Choisissez votre ensemble de données (`chess-games`), puis pour Actions, sélectionnez Supprimer.
6. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/. Supprimez le `databrew-output` dossier et son contenu.

   (Facultatif) Si vous êtes certain de ne plus avoir besoin de votre compartiment Amazon S3, vous pouvez le supprimer.

Connexion aux données avec AWS Glue DataBrew

Dans AWS Glue DataBrew, un ensemble de données représente des données téléchargées à partir d'un fichier ou stockées ailleurs. Par exemple, les données peuvent être stockées dans Amazon S3, dans une source de données JDBC prise en charge ou dans un catalogue de AWS Glue données. Si vous ne chargez pas un fichier directement vers DataBrew, l'ensemble de données contient également des informations sur la manière de DataBrew vous connecter aux données.

Lorsque vous créez votre jeu de données (par exemple,`inventory-dataset`), vous ne saisissez les détails de connexion qu'une seule fois. À partir de ce moment, DataBrew vous pouvez accéder aux données sous-jacentes pour vous. Grâce à cette approche, vous pouvez créer des projets et développer des transformations pour vos données, sans avoir à vous soucier des détails de connexion ou des formats de fichiers.

Rubriques

Types de fichiers pris en charge pour les sources de données

Les exigences suivantes s'appliquent aux fichiers stockés dans Amazon S3 et aux fichiers que vous chargez depuis un disque local. DataBrew prend en charge les formats de fichier suivants : valeur séparée par des virgules (CSV), Microsoft Excel, JSON, ORC et Parquet. Vous pouvez utiliser des fichiers avec une extension non standard ou sans extension s'ils appartiennent à l'un des types pris en charge.

S'il n' DataBrew est pas possible de déduire le type de fichier, assurez-vous de sélectionner vous-même le bon type de fichier (CSV, Excel, JSON, ORC ou Parquet). Les fichiers CSV, JSON, ORC et Parquet compressés sont pris en charge, mais les fichiers CSV et JSON doivent inclure le codec de

compression comme extension de fichier. Si vous importez un dossier, tous les fichiers qu'il contient doivent être du même type.

Les formats de fichiers et les algorithmes de compression pris en charge sont présentés dans le tableau suivant.

> **Note**
>
> Les fichiers CSV, Excel et JSON doivent être codés avec Unicode (UTF-8).

Format	Extension de fichier (facultatif)	Extensions pour les fichiers compressés (obligatoire)

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

classeur Microsoft Excel	`.xlsx`	Aucun support de compression

JSON (document JSON et lignes JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

Format	Extension de fichier (facultatif)	Extensions pour les fichiers compressés (obligatoire)

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

Connexions prises en charge pour les sources de données et les sorties

Vous pouvez vous connecter aux sources de données suivantes pour les tâches de DataBrew recette. Il s'agit notamment de toute source de données qui n'est pas un fichier vers lequel vous importez directement. DataBrew La source de données que vous utilisez peut être appelée base de données, entrepôt de données ou autre. Nous désignons tous les fournisseurs de données par le terme « sources de données » ou « connexions ».

Vous pouvez créer un ensemble de données en utilisant l'une des sources de données suivantes.

Vous pouvez également utiliser les bases de données Amazon S3 ou JDBC prises en charge par Amazon RDS pour la sortie des tâches de DataBrew recette.AWS Glue Data Catalog Amazon AppFlow et les magasins de données AWS Data Exchange ne sont pas pris en charge pour la sortie des tâches de DataBrew recette.

- Amazon S3

  Vous pouvez utiliser S3 pour stocker et protéger n'importe quel volume de données. Pour créer un ensemble de données, vous devez spécifier une URL S3 DataBrew permettant d'accéder à un fichier de données, par exemple : `s3://your-bucket-name/inventory-data.csv`

  DataBrew peut également lire tous les fichiers d'un dossier S3, ce qui signifie que vous pouvez créer un ensemble de données qui couvre plusieurs fichiers. Pour ce faire, spécifiez une URL S3 sous cette forme :`s3://your-bucket-name/your-folder-name/`.

  DataBrew prend uniquement en charge les classes de stockage Amazon S3 suivantes : Standard, Reduced Redundancy et S3 One. Standard-IA Zone-IA DataBrew ignore les fichiers appartenant à d'autres classes de stockage. DataBrew ignore également les fichiers vides (fichiers contenant

0 octet). Pour plus d'informations sur les classes de stockage Amazon S3, consultez la section Utilisation des classes de stockage Amazon S3 dans le guide de l'utilisateur de la console Amazon S3.

- AWS Glue Data Catalog

  Vous pouvez utiliser le catalogue de données pour définir des références aux données stockées dans le AWS cloud. Le catalogue de données vous permet de créer des connexions à des tables individuelles dans les services suivants :

  - Catalogue de données Amazon S3
  - Catalogue de données Amazon Redshift
  - Catalogue de données Amazon RDS
  - AWS Glue

  DataBrew peut également lire tous les fichiers d'un dossier Amazon S3, ce qui signifie que vous pouvez créer un ensemble de données qui couvre plusieurs fichiers. Pour ce faire, spécifiez une URL Amazon S3 sous la forme suivante : `s3://your-bucket-name/your-folder-name/`

  Pour être utilisées avec DataBrew, les tables Amazon S3 définies dans le AWS Glue Data Catalog, doivent être associées à une propriété de table appelée `aclassification`, qui identifie le format des données comme `csvjson`, ou`parquet`, et `typeOfData` comme`file`. Si la propriété de table n'a pas été ajoutée lors de la création de la table, vous pouvez l'ajouter à l'aide de la AWS Glue console.

  DataBrew prend uniquement en charge les classes de stockage Amazon S3 Standard, Reduced Redundancy et S3 One. Standard-IA Zone-IA DataBrew ignore les fichiers appartenant à d'autres classes de stockage. DataBrew ignore également les fichiers vides (fichiers contenant 0 octet). Pour plus d'informations sur les classes de stockage Amazon S3, consultez la section Utilisation des classes de stockage Amazon S3 dans le guide de l'utilisateur de la console Amazon S3.

  DataBrew peut également accéder aux tables AWS Glue Data Catalog S3 depuis d'autres comptes si une politique de ressources appropriée est créée. Vous pouvez créer une politique dans la AWS Glue console sous l'onglet Paramètres sous Catalogue de données. Voici un exemple de politique spécifique à un célibataire Région AWS.

> **Warning**
>
> Il s'agit d'une politique de ressources très permissive qui accorde un accès *$ACCOUNT_TO* illimité au catalogue de données de. *$ACCOUNT_FROM* Dans la plupart des cas, nous vous recommandons de limiter votre politique de ressources à des catalogues ou à des tables spécifiques. Pour plus d'informations, consultez les politiques relatives aux AWS Glue ressources pour le contrôle d'accès dans le Guide du AWS Glue développeur.

Dans certains cas, vous souhaiterez peut-être créer un projet ou exécuter une tâche AWS Glue DataBrew dans *$ACCOUNT_TO* une table AWS Glue Data Catalog S3 *$ACCOUNT_FROM* pointant vers un emplacement S3 également intégré*$ACCOUNT_FROM*. Dans de tels cas, le rôle IAM utilisé lors de la création du projet et de la tâche *$ACCOUNT_TO* doit être autorisé à répertorier et à obtenir des objets se trouvant dans cet emplacement S3. *$ACCOUNT_FROM* Pour plus d'informations, consultez la section Octroi d'un accès multicompte dans le guide du AWS Glue développeur.

- Données connectées à l'aide de pilotes JDBC

  Vous pouvez créer un ensemble de données en vous connectant aux données à l'aide d'un pilote JDBC compatible. Pour de plus amples informations, veuillez consulter Utilisation de pilotes avec AWS Glue DataBrew.

  DataBrew prend officiellement en charge les sources de données suivantes à l'aide de Java Database Connectivity (JDBC) :

  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Amazon Redshift
  - Connecteur Snowflake pour Spark

  Les sources de données peuvent être situées à n'importe quel endroit depuis lequel vous pouvez vous y connecter DataBrew. Cette liste inclut uniquement les connexions JDBC que nous avons testées et que nous pouvons donc prendre en charge.

Les sources de données Amazon Redshift et Snowflake Connector pour Spark peuvent être connectées de l'une des manières suivantes :

- Avec un nom de table.
- Avec une requête SQL qui couvre plusieurs tables et opérations.

Les requêtes SQL sont exécutées lorsque vous démarrez un projet ou que vous exécutez une tâche.

Pour vous connecter à des données qui nécessitent un pilote JDBC non répertorié, assurez-vous que le pilote est compatible avec le JDK 8. Pour utiliser le pilote, stockez-le dans S3 dans un compartiment auquel vous pouvez accéder avec votre rôle IAM pour DataBrew. Pointez ensuite votre ensemble de données sur le fichier du pilote. Pour de plus amples informations, veuillez consulter Utilisation de pilotes avec AWS Glue DataBrew.

Exemple de requête pour un SQL-based ensemble de données :

```
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
```

Limites du SQL personnalisé

Si vous utilisez une connexion JDBC pour accéder aux données d'un DataBrew ensemble de données, gardez à l'esprit les points suivants :

- AWS Glue DataBrew ne valide pas le code SQL personnalisé que vous fournissez dans le cadre de la création du jeu de données. La requête SQL est exécutée lorsque vous lancez un projet ou une tâche. DataBrew prend la requête que vous fournissez et la transmet au moteur de base de données à l'aide des pilotes JDBC par défaut ou fournis.
- Un ensemble de données créé avec une requête non valide échouera s'il est utilisé dans un projet ou une tâche. Validez votre requête avant de créer l'ensemble de données.

- La fonctionnalité Validate SQL n'est disponible que pour les sources Redshift-based de données Amazon.
- Si vous souhaitez utiliser un ensemble de données dans un projet, limitez le temps d'exécution des requêtes SQL à moins de trois minutes afin d'éviter un délai d'attente lors du chargement du projet. Vérifiez l'exécution de la requête avant de créer un projet.

- Amazon AppFlow

  À l'aide d'Amazon AppFlow, vous pouvez transférer des données vers Amazon S3 à partir d'applications tierces Software-as-a-Service (SaaS) telles que Salesforce, Zendesk, Slack et. ServiceNow Vous pouvez ensuite utiliser les données pour créer un DataBrew ensemble de données.

  Dans Amazon AppFlow, vous créez une connexion et un flux pour transférer des données entre votre application tierce et une application de destination. Lorsque vous utilisez Amazon AppFlow avec DataBrew, assurez-vous que l'application de AppFlow destination Amazon est Amazon S3. Les applications de AppFlow destination Amazon autres qu'Amazon S3 n'apparaissent pas dans la DataBrew console. Pour plus d'informations sur le transfert de données depuis votre application tierce et la création de AppFlow connexions et de flux Amazon, consultez la AppFlow documentation Amazon.

  Lorsque vous sélectionnez Connect new dataset dans l'onglet Datasets de DataBrew et AppFlow que vous cliquez sur Amazon AppFlow, tous les flux d'Amazon configurés avec Amazon S3 comme application de destination s'affichent. Pour utiliser les données d'un flux pour votre ensemble de données, choisissez ce flux.

  Choisissez Créer un flux, Gérer les flux et Afficher les détails d'Amazon AppFlow dans la DataBrew console pour ouvrir la AppFlow console Amazon afin que vous puissiez effectuer ces tâches.

  Après avoir créé un ensemble de données à partir d'Amazon AppFlow, vous pouvez exécuter le flux et consulter les détails de la dernière exécution du flux lorsque vous consultez les détails du jeu de données ou les détails des tâches. Lorsque vous exécutez le flux DataBrew, le jeu de données est mis à jour dans S3 et est prêt à être utilisé dans DataBrew.

  Les situations suivantes peuvent survenir lorsque vous sélectionnez un AppFlow flux Amazon dans la DataBrew console pour créer un ensemble de données :

  - Les données n'ont pas été agrégées : si le déclencheur du flux est Exécuté à la demande ou s'il est exécuté selon le calendrier avec transfert complet des données, assurez-vous d'agréger les données du flux avant de les utiliser pour créer un DataBrew ensemble de données. L'agrégation

du flux combine tous les enregistrements du flux dans un seul fichier. Les flux dont le type de déclencheur est Exécuter selon le calendrier avec transfert de données incrémentiel ou Exécuter selon un événement ne nécessitent pas d'agrégation. Pour agréger les données dans Amazon AppFlow, choisissez Modifier la configuration du flux > Détails de la destination > Paramètres supplémentaires > Préférence de transfert de données.

- Le flux n'a pas été exécuté : si le statut d'exécution d'un flux est vide, cela signifie l'une des choses suivantes :
  - Si le déclencheur pour exécuter le flux est Exécuter à la demande, le flux n'a pas encore été exécuté.
  - Si le déclencheur de l'exécution du flux est Run on event, l'événement déclencheur ne s'est pas encore produit.
  - Si le déclencheur pour exécuter le flux est Exécuter selon le calendrier, aucune exécution planifiée n'a encore eu lieu.

  Avant de créer un ensemble de données avec un flux, choisissez Run flow pour ce flux.

  Pour plus d'informations, consultez Amazon AppFlow flows dans le guide de AppFlow l'utilisateur Amazon.
- AWS Data Exchange

  Vous pouvez choisir parmi des centaines de sources de données tierces disponibles dans AWS Data Exchange. En vous abonnant à ces sources de données, vous obtenez la version la plus récente des données.

  Pour créer un ensemble de données, vous devez spécifier le nom d'un produit de AWS Data Exchange données auquel vous êtes abonné et que vous êtes autorisé à utiliser.

Utilisation de jeux de données dans AWS Glue DataBrew

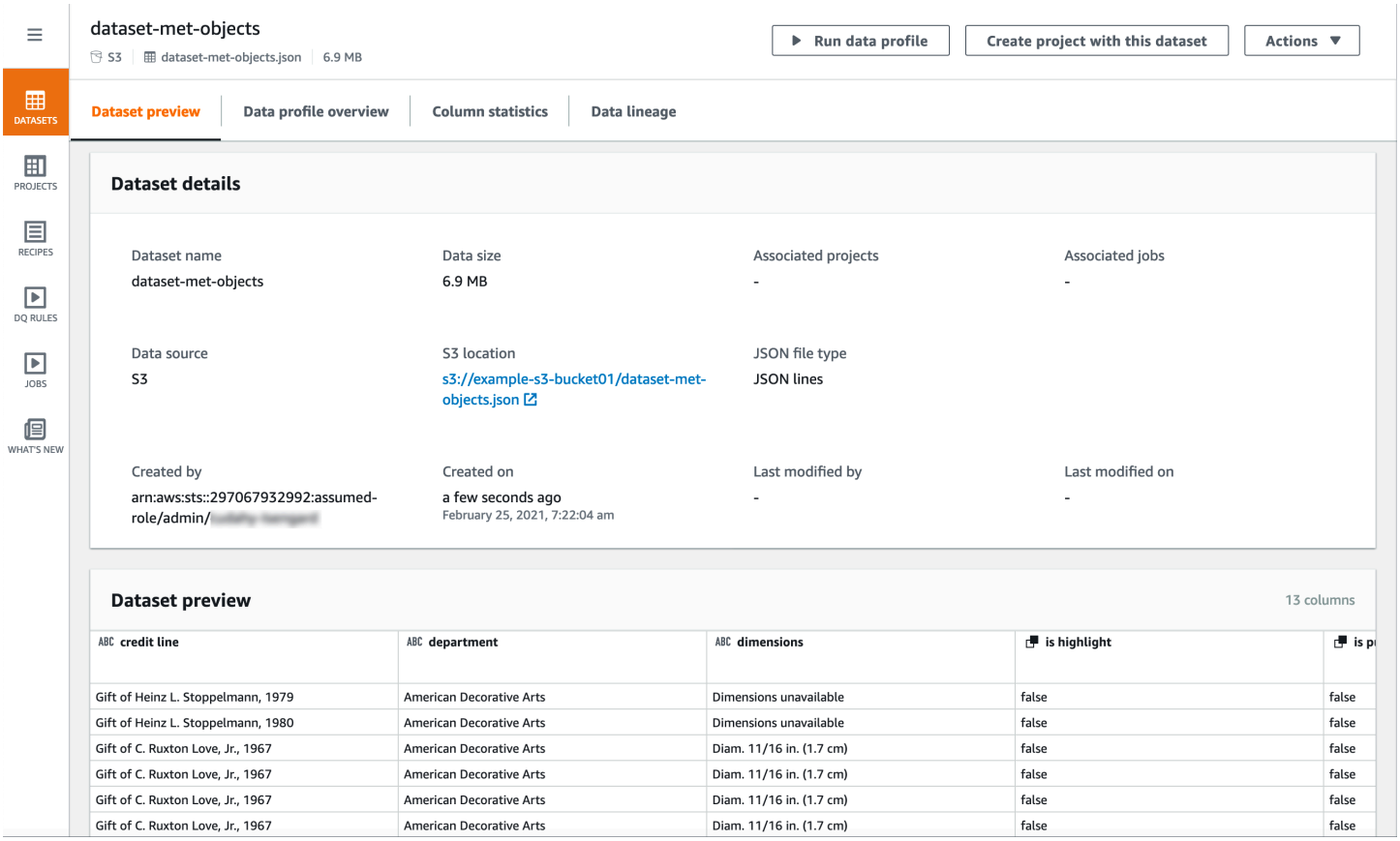
Pour afficher la liste de vos ensembles de données dans la DataBrew console, choisissez DATASET sur la gauche. Sur la page des ensembles de données, vous pouvez consulter les informations détaillées de chaque jeu de données en cliquant sur son nom ou en choisissant Actions, Modifier dans le menu contextuel.

Pour créer un nouvel ensemble de données, choisissez DATASET, Connect new dataset. Les différentes sources de données ont des paramètres de connexion différents, et vous les entrez pour DataBrew pouvoir vous connecter. Lorsque vous enregistrez votre connexion et que vous

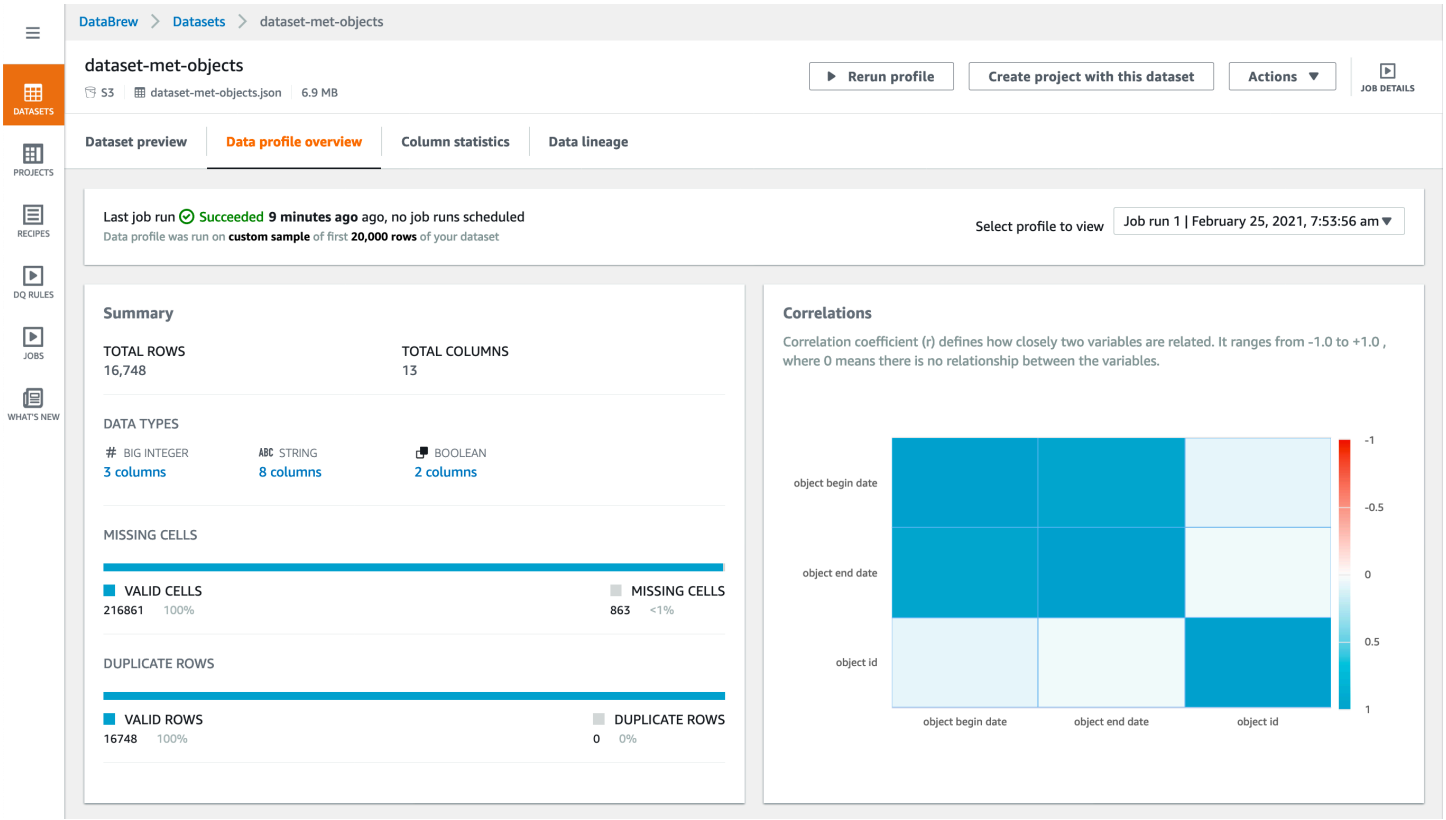
choisissez Create dataset, vous DataBrew vous connectez à vos données et commencez à charger les données. Pour de plus amples informations, veuillez consulter Connexion à vos données.

La page du jeu de données contient les éléments suivants pour vous aider à explorer vos données.

Aperçu du jeu de données : dans cet onglet, vous trouverez des informations de connexion pour le jeu de données et un aperçu de la structure globale du jeu de données, comme indiqué ci-dessous.

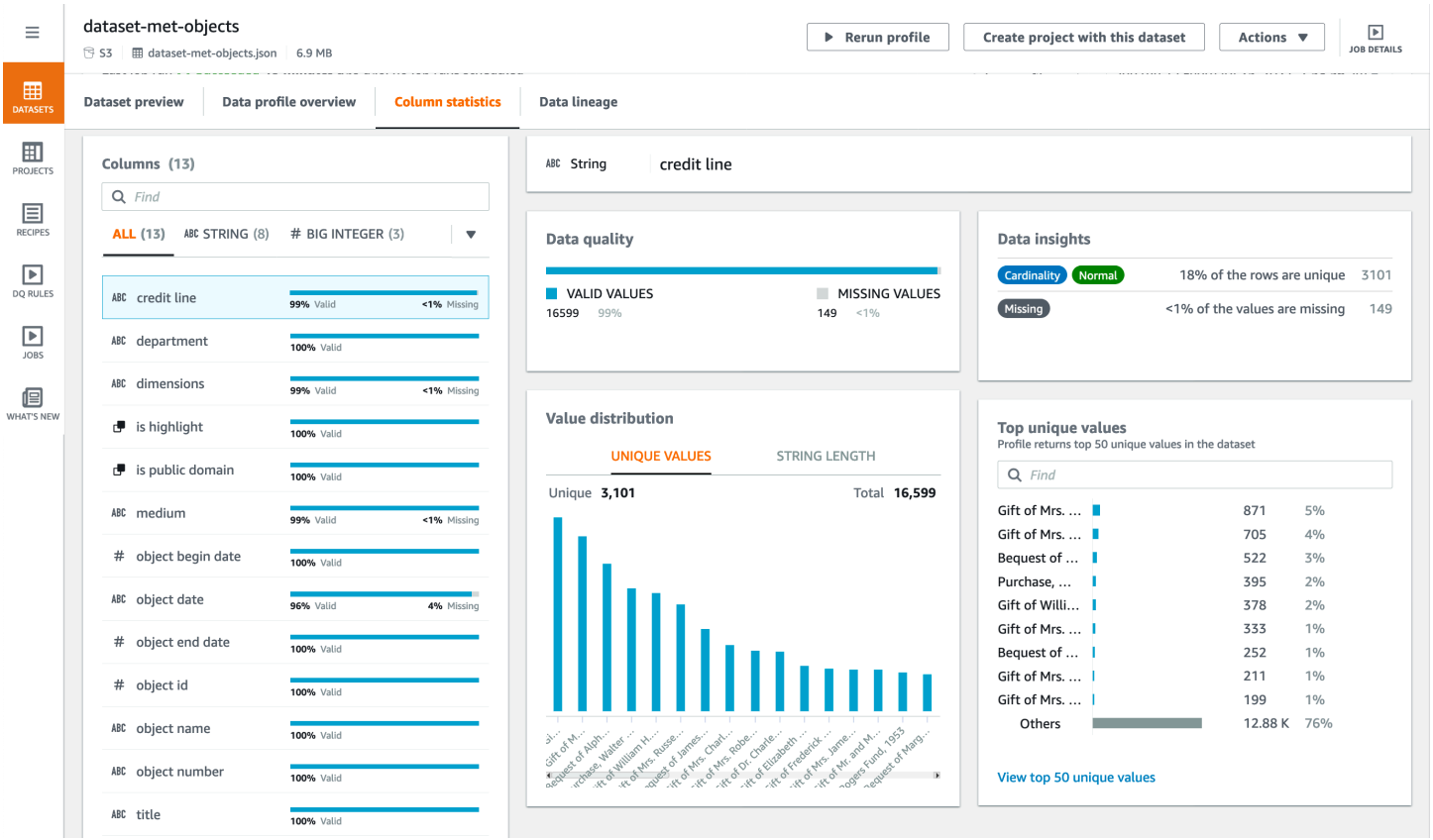


Vue d'ensemble du profil de données — Dans cet onglet, vous pouvez trouver un profil de données graphique contenant des statistiques et des données volumétriques pour votre ensemble de données, comme indiqué ci-dessous.

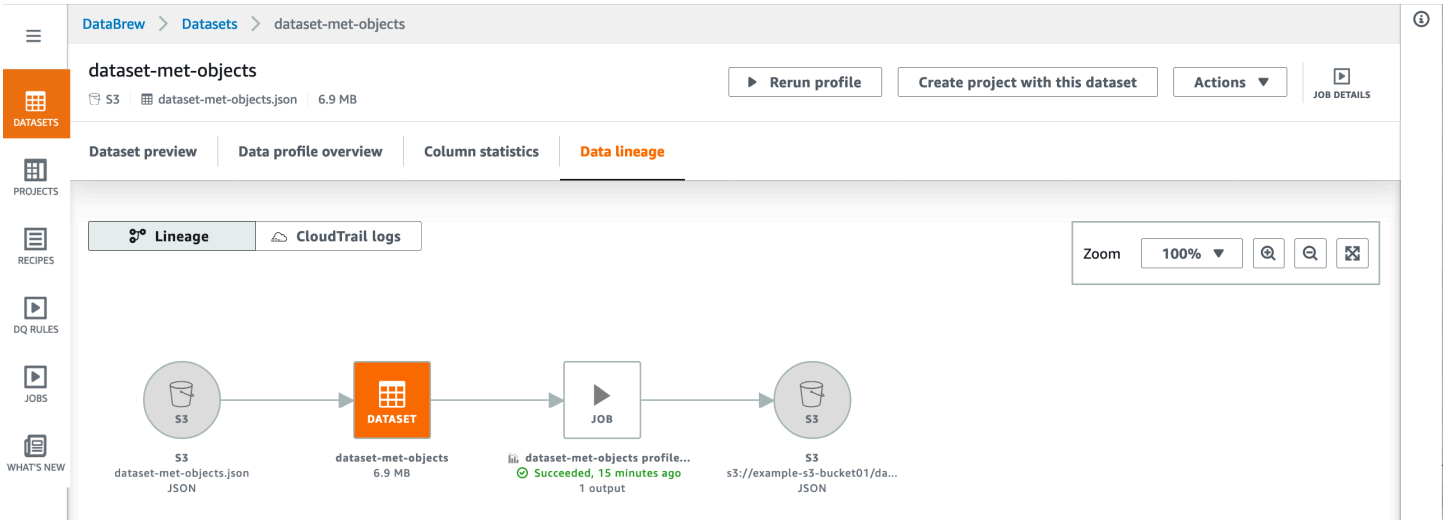


> **Note**
>
> Pour créer un profil de données, exécutez une tâche DataBrew de profilage sur votre ensemble de données. Pour plus d'informations sur la procédure à utiliser, consultez Étape 5 : Création d'un profil de données.

Statistiques des colonnes : dans cet onglet, vous pouvez trouver des statistiques détaillées sur chaque colonne de votre ensemble de données, comme indiqué ci-dessous.



Lignage des données — Cet onglet affiche une représentation graphique de la façon dont votre ensemble de données a été créé et de la manière dont il est utilisé DataBrew, comme indiqué ci-dessous.



Rubriques

- [Suppression d’un jeu de données]

Suppression d’un jeu de données

Si vous n'avez plus besoin d'un jeu de données, vous pouvez le supprimer. La suppression d'un ensemble de données n'a aucune incidence sur la source de données sous-jacente. Il supprime simplement les informations DataBrew utilisées pour accéder à la source de données.

Vous ne pouvez pas supprimer un ensemble de données si d'autres DataBrew ressources en dépendent. Par exemple, si vous avez actuellement un DataBrew projet qui utilise le jeu de données, supprimez-le d'abord avant de supprimer le jeu de données.

Pour supprimer un ensemble de données, choisissez Dataset dans le volet de navigation. Choisissez le jeu de données que vous souhaitez supprimer, puis dans Actions, sélectionnez Supprimer.

Connexion à vos données

Pour plus d'informations sur la connexion aux sources de données suivantes, choisissez la section qui vous concerne.

- AWS Glue Data Catalog— Vous pouvez utiliser le catalogue de données pour définir des références à des objets de données stockés dans le AWS cloud, notamment aux services suivants :
  - Amazon Redshift
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS for MySQL
  - Amazon RDS pour PostgreSQL

  DataBrew reconnaît toutes les autorisations de Lake Formation qui ont été appliquées aux ressources du catalogue de données, de sorte que DataBrew les utilisateurs ne peuvent accéder à ces ressources que s'ils sont autorisés.

  Pour créer un jeu de données, vous devez spécifier un nom de base de données de catalogue de données et un nom de table. DataBrew s'occupe des autres détails de connexion.
- AWS Data Exchange : vous pouvez choisir parmi des centaines de sources de données tierces disponibles dans AWS Data Exchange. En vous abonnant à ces sources de données, vous disposez toujours de la version la plus récente des données.

Pour créer un ensemble de données, vous devez spécifier le nom d'un produit de données Data Exchange auquel vous êtes abonné ou que vous êtes autorisé à utiliser.

- Connexions au pilote JDBC : vous pouvez créer un ensemble de données en vous connectant DataBrew à une source de JDBC-compatible données. DataBrew prend en charge la connexion aux sources suivantes via JDBC :
  - Amazon Redshift
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Snowflake

Rubriques

Utilisation de pilotes avec AWS Glue DataBrew

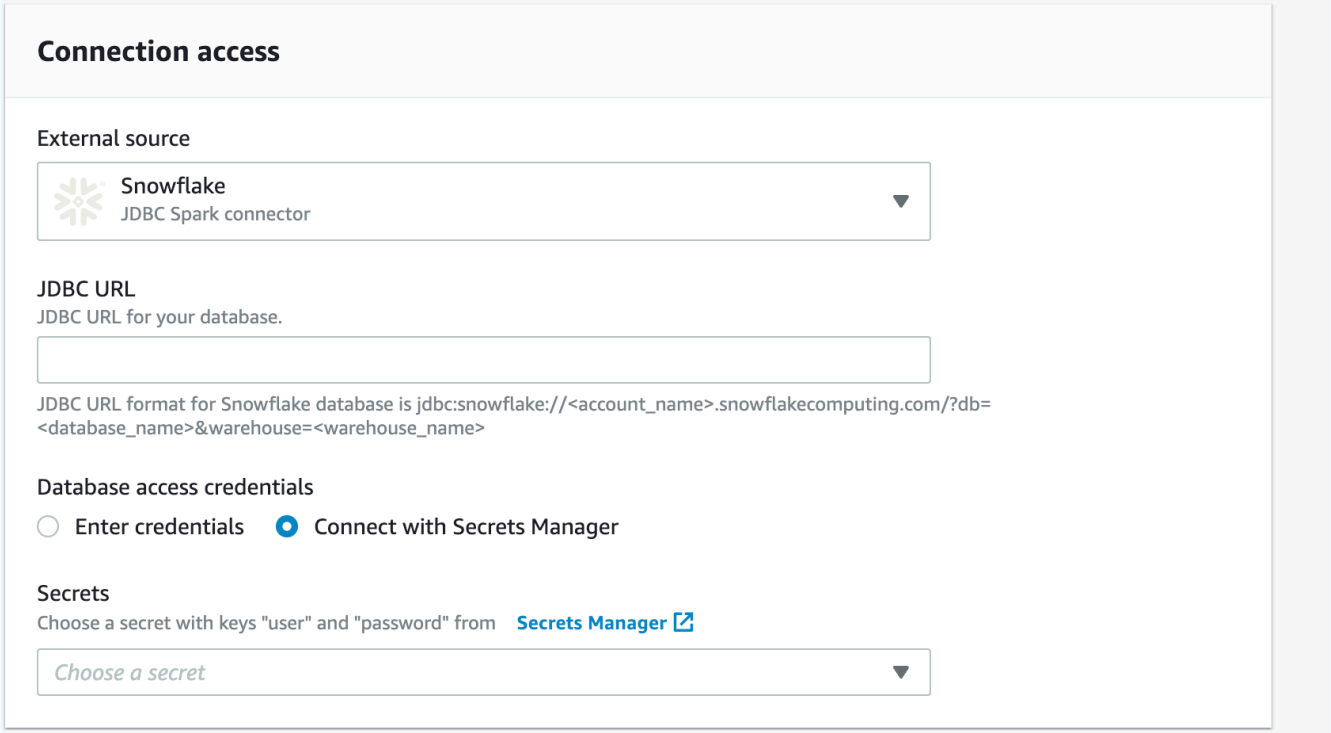
Un pilote de base de données est un fichier ou une URL qui implémente un protocole de connexion à une base de données, par exemple Java Database Connectivity (JDBC). Le pilote fonctionne comme un adaptateur ou un traducteur entre un système de gestion de base de données (DBMS) spécifique et un autre système.

Dans ce cas, il permet AWS Glue DataBrew de se connecter à vos données. Vous pouvez ensuite accéder à un objet de base de données, tel qu'une table ou une vue, à partir d'une source de données prise en charge. La source de données que vous utilisez peut être appelée base de données, entrepôt de données ou autre. Toutefois, dans le cadre de cette documentation, nous désignons tous les fournisseurs de données par le terme « sources de données » ou « connexions ».

Pour utiliser un pilote JDBC ou un fichier JAR, téléchargez le ou les fichiers dont vous avez besoin et placez-les dans un compartiment S3. Le rôle IAM que vous utilisez pour accéder aux données doit disposer d'autorisations de lecture pour les deux fichiers de pilote.

Note

With AWS Glue4.0, la connexion à Snowflake en tant que source de données est prise en charge de manière native. Il n'est pas nécessaire de fournir des `jar` fichiers personnalisés. Dans AWS Glue DataBrew, choisissez Snowflake comme connexion source externe et fournissez l'URL de votre instance Snowflake. L'URL utilisera un nom d'hôte sous la forme `https://account_identifier.snowflakecomputing.com`.
Fournissez les informations d'identification d'accès aux données, le nom de la base de données Snowflake et le nom du schéma Snowflake. De plus, si votre utilisateur Snowflake n'a pas défini d'entrepôt par défaut, vous devrez fournir un nom d'entrepôt.
Les connexions Snowflake utilisent un AWS Secrets Manager secret pour fournir des informations d'identification. Votre projet et vos rôles professionnels doivent être autorisés à lire ce secret.



Pour utiliser des pilotes avec DataBrew

1. Découvrez la version de votre source de données utilisée à l'aide de la méthode fournie par le produit.
2. Trouvez la dernière version des connecteurs et du pilote requis. Vous pouvez trouver ces informations sur le site Web du fournisseur de données.

3. Téléchargez la version requise des fichiers JDBC. Ils sont généralement stockés sous forme de fichiers Java ArchiVE (.JAR).
4. Téléchargez les pilotes depuis la console vers votre compartiment S3 ou fournissez le chemin S3 vers vos fichiers .JAR.
5. Entrez les détails de connexion de base, par exemple la classe, l'instance, etc.
6. Entrez toutes les informations de configuration supplémentaires dont votre source de données a besoin, par exemple les informations relatives au cloud privé virtuel (VPC).

Pilotes JDBC pris en charge

Produit (langue française non garantie)	Version prise en charge	Instructions relatives au pilote et téléchargements	Requêtes SQL prises en charge
Microsoft S	v6.x ou supérieur	Pilote Microsoft JDBC pour SQL Server	Non pris en charge
MySQL	v5.1 ou supérieur	Connecteurs MySQL	Non pris en charge
Oracle	v11.2 ou supérieur	Téléchargements d'Oracle JDBC	Non pris en charge
PostgreSQL	v4.2.x ou supérieur	pilote JDBC pour PostgreSQL	Non pris en charge
Amazon Re	v4.1 ou supérieur	Connexion à Amazon Redshift avec JDBC	Pris en charge

Produit (langue française non garantie)	Version prise en charge	Instructions relatives au pilote et téléchargements	Requêtes SQL prises en charge
Snowflake	Pour voir votre version de Snowflake, utilisez CURRENT_VERSION comme décrit dans la documentation de Snowflake.	Pour vous connecter à Snowflake, vous avez besoin des deux éléments suivants : • Pilote JDBC Snowflake • Connecteur Snowflake pour Spark	Pris en charge

Pour vous connecter à des bases de données ou à des entrepôts de données qui nécessitent une version du pilote différente de celle prise en charge de DataBrew manière native, vous pouvez fournir le pilote JDBC de votre choix. Le pilote doit être compatible avec JDK 8 ou Java 8. Pour savoir comment trouver la dernière version du pilote pour votre base de données, consultezUtilisation de pilotes avec AWS Glue DataBrew.

Connexion aux données d'un fichier texte avec DataBrew

Vous pouvez configurer les options de format suivantes pour les fichiers d'entrée DataBrew compatibles :

- Comma-separated fichiers de valeurs (CSV)
  - Délimiteurs

Le délimiteur par défaut est une virgule pour les fichiers .csv. Si votre fichier utilise un autre délimiteur, choisissez le délimiteur pour le délimiteur CSV dans la section Configurations supplémentaires lorsque vous créez votre jeu de données. Les délimiteurs suivants sont pris en charge pour les fichiers .csv :

- Virgule (,)
- Côlon (:)
- Semi-colon (;)
- Pipe (|)
- Tabulation (\t)
- Caret (^)
- Barre oblique inverse (\)
- Espace

- Valeurs d'en-tête de colonne

  Votre fichier CSV peut inclure une ligne d'en-tête comme première ligne du fichier. Si ce n'est pas le cas, DataBrew crée une ligne d'en-tête pour vous.

  - Si votre fichier CSV inclut une ligne d'en-tête, choisissez Traiter la première ligne comme en-tête. Dans ce cas, la première ligne de votre fichier CSV est considérée comme contenant les valeurs d'en-tête de colonne.
  - Si votre fichier CSV ne contient pas de ligne d'en-tête, choisissez Ajouter un en-tête par défaut. Si vous le faites, DataBrew crée une ligne d'en-tête pour le fichier et ne traitez pas votre première ligne de données comme contenant des valeurs d'en-tête. Les en-têtes DataBrew créés se composent d'un trait de soulignement et d'un chiffre pour chaque colonne du fichier, au format `Column_1Column_2`,`Column_3`,, etc.

- fichiers JSON

  DataBrew prend en charge deux formats pour les fichiers JSON, les lignes JSON et les documents JSON. Les fichiers JSON Lines contiennent une ligne par ligne. Dans les fichiers de documents JSON, toutes les lignes sont contenues dans une structure JSON unique ou dans un tableau. Vous pouvez spécifier votre type de fichier JSON dans la section Configurations supplémentaires lorsque vous créez un jeu de données JSON. Le format par défaut est JSON Lines.

- fichiers Excel

  Les règles suivantes s'appliquent aux feuilles Excel dans DataBrew :

- Chargement de feuilles Excel

  Par défaut, DataBrew charge la première feuille de votre fichier Excel. Toutefois, vous pouvez spécifier un numéro ou un nom de feuille différent dans la section Configurations supplémentaires lorsque vous créez un jeu de données Excel.

- Valeurs d'en-tête de colonne

  Vos feuilles Excel peuvent inclure une ligne d'en-tête comme première ligne du fichier, mais si ce n'est pas le cas, elles DataBrew créeront une ligne d'en-tête pour vous.

  - Si vos feuilles Excel incluent une ligne d'en-tête, choisissez Traiter la première ligne comme en-tête. Dans ce cas, la première ligne de vos feuilles Excel est considérée comme contenant les valeurs d'en-tête de colonne.
  - Si votre fichier Excel ne contient pas de ligne d'en-tête, choisissez Ajouter un en-tête par défaut. Ce faisant, vous spécifiez que vous DataBrew devez créer une ligne d'en-tête pour le fichier et ne pas traiter votre première ligne de données comme contenant des valeurs d'en-tête. Les en-têtes DataBrew créés se composent d'un trait de soulignement et d'un chiffre pour chaque colonne du fichier, au format `Column_1Column_2,Column_3,`, etc.

Connexion de données dans plusieurs fichiers dans Amazon S3

Avec la DataBrew console, vous pouvez parcourir les compartiments et les dossiers Amazon S3 et choisir un fichier pour votre ensemble de données. Cependant, un ensemble de données n'a pas besoin d'être limité à un seul fichier.

Supposons que vous disposiez d'un compartiment S3 nommé `my-databrew-bucket` contenant un dossier nommé`databrew-input`. Dans ce dossier, supposons que vous avez plusieurs fichiers JSON, tous dotés du même format de fichier et de la même extension de `.json` fichier. Sur la console, vous pouvez spécifier l'URL source de`s3://my-databrew-bucket/databrew-input/`. Sur la DataBrew console, vous pouvez ensuite choisir ce dossier. Votre ensemble de données comprend tous les fichiers JSON de ce dossier.

DataBrew peut traiter tous les fichiers d'un dossier S3, mais uniquement si les conditions suivantes sont remplies :

- Tous les fichiers du dossier ont le même format.
- Tous les fichiers du dossier ont la même extension de fichier.

Pour plus d'informations sur les formats de fichiers et les extensions pris en charge, consultezDataBrew input formats.

Schémas lors de l'utilisation de plusieurs fichiers en tant que jeu de données

Lorsque vous utilisez plusieurs fichiers en tant que DataBrew jeu de données, les schémas doivent être identiques dans tous les fichiers. Sinon, l'espace de travail du projet essaie automatiquement de choisir l'un des schémas parmi les multiples fichiers et essaie de conformer le reste des fichiers du jeu de données à ce schéma. En raison de ce comportement, l'affichage affiché dans Project Workspace est irrégulier et, par conséquent, le résultat du travail sera également irrégulier.

Si vos fichiers doivent avoir des schémas différents, vous devez créer plusieurs ensembles de données et les profiler séparément.

Utilisation de chemins paramétrés pour Amazon S3

Dans certains cas, vous souhaiterez peut-être créer un ensemble de données contenant des fichiers respectant une certaine convention de dénomination, ou un ensemble de données pouvant couvrir plusieurs dossiers Amazon S3. Vous pouvez également réutiliser le même ensemble de données pour des données structurées de manière identique qui sont générées périodiquement dans un emplacement S3 avec un chemin qui dépend de certains paramètres. Un chemin nommé d'après la date de production des données en est un exemple.

DataBrew prend en charge cette approche avec des chemins S3 paramétrés. Un chemin paramétré est une URL Amazon S3 contenant des expressions régulières ou des paramètres de chemin personnalisés, ou les deux.

Définition d'un ensemble de données avec un chemin S3 à l'aide d'expressions régulières

Les expressions régulières dans le chemin peuvent être utiles pour faire correspondre plusieurs fichiers d'un ou de plusieurs dossiers tout en filtrant les fichiers non liés présents dans ces dossiers.

Voici quelques exemples :

- Définissez un ensemble de données comprenant tous les fichiers JSON d'un dossier dont le nom commence par`invoice`.

- Définissez un ensemble de données incluant tous les fichiers `2020` dans des dossiers portant leur nom.

Vous pouvez implémenter ce type d'approche en utilisant des expressions régulières dans le chemin S3 d'un ensemble de données. Ces expressions régulières peuvent remplacer n'importe quelle sous-chaîne de la clé de l'URL S3 (mais pas le nom du compartiment).

À titre d'exemple de clé dans une URL S3, reportez-vous à ce qui suit. `my-bucket`Voici le nom du compartiment, US East (Ohio) la AWS région et `puppy.png` le nom clé.

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

Dans un chemin S3 paramétré, tous les caractères entre deux crochets (<et>) sont traités comme des expressions régulières. Voici deux exemples :

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json`correspond à tous les fichiers nommés`data.json`, dans tous les sous-dossiers `databrew-input` dont le nom commence `invoice` par.
- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/`fait correspondre tous les fichiers des dossiers dont `2020` le nom est indiqué.

Dans ces exemples, `.*` correspond à zéro caractère ou plus.

> **Note**
>
> Vous ne pouvez utiliser des expressions régulières que dans la partie clé du chemin S3, c'est-à-dire la partie qui suit le nom du compartiment. Donc, `s3://my-databrew-bucket/<.*>-input/` c'est valide, mais ne l'`s3://my-<.*>-bucket/<.*>-input/`est pas.

Nous vous recommandons de tester vos expressions régulières pour vous assurer qu'elles correspondent uniquement aux URL S3 que vous souhaitez, et non à celles que vous ne souhaitez pas.

Voici d'autres exemples d'expressions régulières :

- `<\d{2}>`correspond à une chaîne composée exactement de deux chiffres consécutifs, par exemple `07` ou`03`, mais non`1a2`.

- `<[a-z]+.*>`correspond à une chaîne qui commence par une ou plusieurs lettres latines minuscules et qui est suivie de zéro ou de plusieurs autres caractères. Un exemple est `a3abc/def`, ou`a-z`, mais non`A2`.
- `<[^/]+>`correspond à une chaîne contenant tous les caractères à l'exception d'une barre oblique (/). Dans une URL S3, des barres obliques sont utilisées pour séparer les dossiers du chemin.
- `<.*=.*>`correspond à une chaîne contenant un signe égal (=), par exemple`month=02`, ou `abc/day=2=10`, mais non`test`.
- `<\d.*\d>`correspond à une chaîne qui commence et se termine par un chiffre et qui peut contenir n'importe quel autre caractère entre les chiffres`1abc2`, par exemple`01-02-03`, ou`2020/Jul/21`, mais pas`123a`.

Définition d'un ensemble de données avec un chemin S3 à l'aide de paramètres personnalisés

La définition d'un jeu de données paramétré à l'aide de paramètres personnalisés présente des avantages par rapport à l'utilisation d'expressions régulières lorsque vous souhaitez fournir des paramètres pour un emplacement S3 :

- Vous pouvez obtenir les mêmes résultats qu'avec une expression régulière, sans avoir besoin de connaître la syntaxe des expressions régulières. Vous pouvez définir des paramètres en utilisant des termes courants tels que « commence par » et « contient ».
- Lorsque vous définissez un jeu de données dynamique à l'aide des paramètres du chemin, vous pouvez inclure une plage de temps dans votre définition, telle que « le mois dernier » ou « les 24 dernières heures ». Ainsi, la définition de votre jeu de données sera utilisée ultérieurement avec les nouvelles données entrantes.

Voici quelques exemples de situations dans lesquelles vous souhaiterez peut-être utiliser des ensembles de données dynamiques :

- Pour connecter plusieurs fichiers partitionnés en fonction de la date de dernière mise à jour ou d'autres attributs significatifs en un seul jeu de données. Vous pouvez ensuite capturer ces attributs de partition sous forme de colonnes supplémentaires dans un ensemble de données.
- Limiter les fichiers d'un ensemble de données aux emplacements S3 qui répondent à certaines conditions. Supposons, par exemple, que votre chemin S3 contienne des dossiers basés sur des dates tels que`folder/2021/04/01/`. Dans ce cas, vous pouvez paramétrer la date et la limiter

à une certaine plage, comme « entre le 1er mars 2021 et le 1er avril 2021 » ou « La semaine dernière ».

Pour définir un chemin à l'aide de paramètres, définissez les paramètres et ajoutez-les à votre chemin en utilisant le format suivant :

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

> **Note**
>
> Comme pour les expressions régulières dans un chemin S3, vous ne pouvez utiliser des paramètres que dans la partie clé du chemin, c'est-à-dire la partie qui suit le nom du compartiment.

Deux champs sont obligatoires dans la définition d'un paramètre, le nom et le type. Le type peut être String, Number ou Date. Les paramètres de type Date doivent avoir une définition du format de date afin de DataBrew pouvoir interpréter et comparer correctement les valeurs de date. Vous pouvez éventuellement définir des conditions de correspondance pour un paramètre. Vous pouvez également choisir d'ajouter les valeurs correspondantes d'un paramètre sous forme de colonne à votre ensemble de données lorsqu'il est chargé par une DataBrew tâche ou une session interactive.

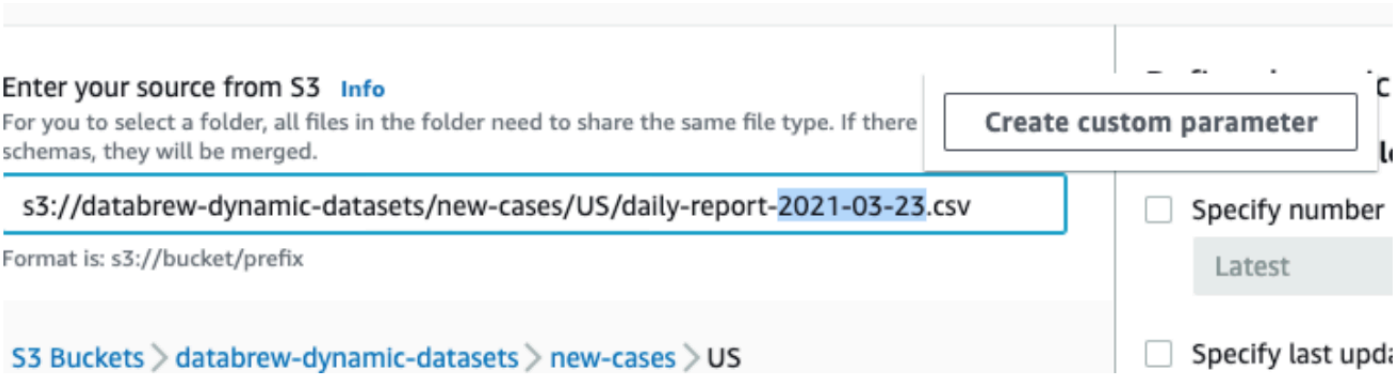
Exemple

Prenons un exemple de définition d'un jeu de données dynamique à l'aide de paramètres dans la DataBrew console. Dans cet exemple, supposons que les données d'entrée soient régulièrement écrites dans un compartiment S3 à l'aide d'emplacements tels que ceux-ci :

- `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`

Il y a deux parties dynamiques ici : un code de pays, comme États-Unis, et une date dans le nom du fichier, comme 2021-03-30. Ici, vous pouvez appliquer la même recette de nettoyage pour tous les fichiers. Supposons que vous souhaitiez effectuer votre travail de nettoyage tous les jours. Voici comment définir un chemin paramétré pour ce scénario :

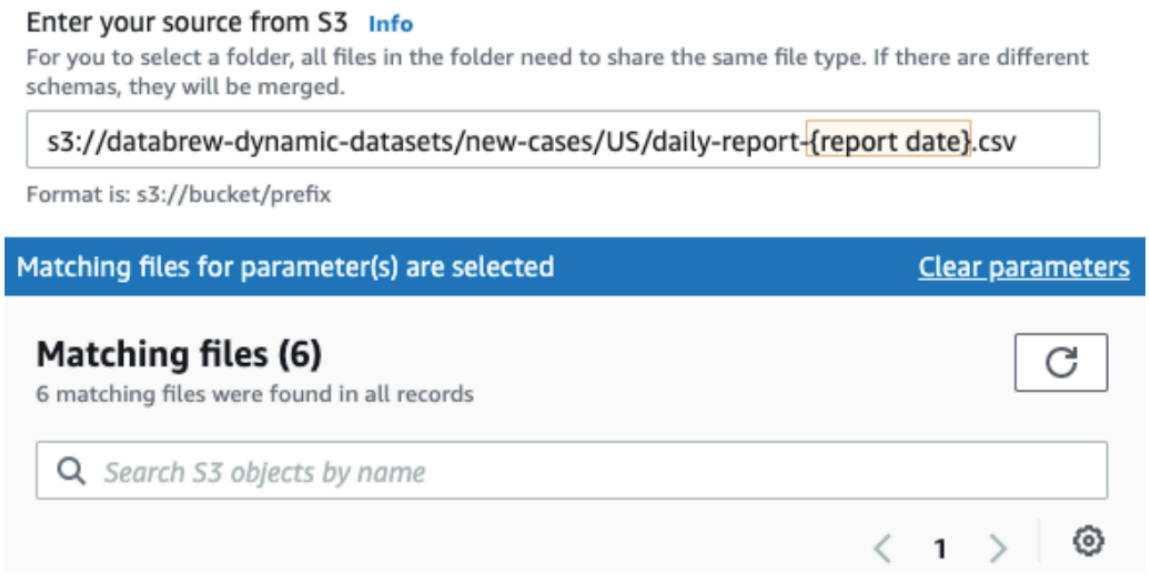
1. Accédez à un fichier spécifique.
2. Sélectionnez ensuite une partie variable, comme une date, et remplacez-la par un paramètre. Dans ce cas, remplacez une date.



3. Ouvrez le menu contextuel (clic droit) pour créer un paramètre personnalisé et définissez ses propriétés :
   - Nom : date du rapport
   - Type : date
   - Format de date : yyyy- MM-dd (sélectionné parmi les formats prédéfinis)
   - Conditions (plage horaire) : 24 heures passées
   - Ajouter en tant que colonne : vrai (coché)

   Conservez les valeurs par défaut des autres champs.
4. Choisissez Créer.

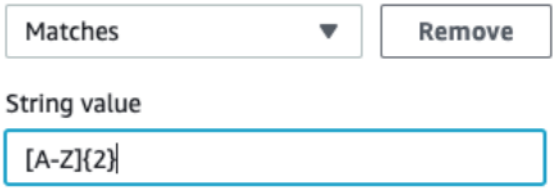
Après cela, le chemin mis à jour s'affiche, comme dans la capture d'écran suivante.



Vous pouvez maintenant faire de même pour le code de pays et le paramétrer comme suit :

- Nom : code du pays
- Type : Chaîne
- Ajouter en tant que colonne : vrai (coché)

Il n'est pas nécessaire de spécifier des conditions si toutes les valeurs sont pertinentes. Dans le `new-cases` dossier, par exemple, nous n'avons que des sous-dossiers contenant des codes de pays, il n'est donc pas nécessaire de définir des conditions. Si vous deviez exclure d'autres dossiers, vous pouvez utiliser la condition suivante.



Cette approche limite les sous-dossiers des nouvelles affaires à deux caractères latins majuscules.

Après ce paramétrage, vous n'avez que les fichiers correspondants dans notre jeu de données et vous pouvez choisir Create Dataset.

> **Note**
>
> Lorsque vous utilisez des plages de temps relatives dans des conditions, celles-ci sont évaluées lors du chargement du jeu de données. Cela est vrai qu'il s'agisse de plages de

temps prédéfinies telles que « 24 dernières heures » ou de plages de temps personnalisées telles que « il y a 5 jours ». Cette approche d'évaluation s'applique que l'ensemble de données soit chargé lors de l'initialisation d'une session interactive ou lors du démarrage d'une tâche.

Une fois que vous avez sélectionné Create Dataset, votre jeu de données dynamique est prêt à être utilisé. Par exemple, vous pouvez d'abord l'utiliser pour créer un projet et définir une recette de nettoyage à l'aide d'une DataBrew session interactive. Vous pouvez ensuite créer une tâche dont l'exécution est planifiée tous les jours. Cette tâche peut appliquer la recette de nettoyage aux fichiers de jeux de données qui répondent aux conditions de vos paramètres au moment du démarrage de la tâche.

Conditions prises en charge pour les ensembles de données dynamiques

Vous pouvez utiliser des conditions pour filtrer les fichiers S3 correspondants à l'aide de paramètres ou de l'attribut de date de dernière modification.

Vous trouverez ci-dessous la liste des conditions prises en charge pour chaque type de paramètre.

Conditions utilisées avec les paramètres String

Nom dans le DataBrew SDK	Synonymes du SDK	Nom dans la DataBrew console	Description
est	éq, ==	C'est exactement	La valeur du paramètre est identique à la valeur fournie dans la condition.
n'est pas	pas EQ, ! =	Is not (N’est pas)	La valeur du paramètre n'est pas la même que celle fournie dans la condition.
contient		Contains	La valeur de chaîne du paramètre contient

Nom dans le DataBrew SDK	Synonymes du SDK	Nom dans la DataBrew console	Description
			la valeur fournie dans la condition.
ne contient pas		Ne contient pas	La valeur de chaîne du paramètre ne contient pas la valeur fournie dans la condition.
commence_par		Starts with	La valeur de chaîne du paramètre commence par la valeur fournie dans la condition.
ne commence pas par		Ne commence pas par	La valeur de chaîne du paramètre ne commence pas par la valeur fournie dans la condition.
se termine_par		Termine par	La valeur de chaîne du paramètre se termine par la valeur fournie dans la condition.
ne se termine pas par		Ne se termine pas par	La valeur de chaîne du paramètre ne se termine pas par la valeur fournie dans la condition.

Nom dans le DataBrew SDK	Synonymes du SDK	Nom dans la DataBrew console	Description
allumettes		Correspondance	La valeur du paramètre correspon d à l'expression régulière fournie dans la condition.
ne correspond pas		Ne correspond pas	La valeur du paramètre ne correspond pas à l'expression régulière fournie dans la condition.

> **Note**
>
> Toutes les conditions relatives aux paramètres de chaîne utilisent la comparaison entre majuscules et minuscules. Si vous n'êtes pas sûr du cas utilisé dans un chemin S3, vous pouvez utiliser la condition « matches » avec une valeur d'expression régulière commençant par`(?i)`. Cela permet d'effectuer une comparaison qui ne fait pas la distinction majuscules/majuscules.
>
> Supposons, par exemple, que vous souhaitiez que votre paramètre de chaîne commence par`abc`, mais `ABC` que cela `Abc` soit également possible. Dans ce cas, vous pouvez utiliser la condition « correspond » `(?i)^abc` comme valeur de condition.

Conditions utilisées avec les paramètres numériques

Nom dans le DataBrew SDK	Synonymes du SDK	Nom dans la DataBrew console	Description
est	éq, ==	C'est exactement	La valeur du paramètre est identique à la valeur

Nom dans le DataBrew SDK	Synonymes du SDK	Nom dans la DataBrew console	Description
			fournie dans la condition.
n'est pas	pas EQ, ! =	Is not (N’est pas)	La valeur du paramètre n'est pas la même que celle fournie dans la condition.
inférieur_que	lt, <	Inférieur à	La valeur numérique du paramètre est inférieure à la valeur fournie dans la condition.
moins_que_égal	Oui, <=	Inférieur ou égal à	La valeur numérique du paramètre est inférieure ou égale à la valeur fournie dans la condition.
supérieur_que	gt, >	Supérieur à	La valeur numérique du paramètre est supérieure à la valeur fournie dans la condition.
supérieur à l'égal	lit, >=	Supérieur ou égal à	La valeur numérique du paramètre est supérieure ou égale à la valeur fournie dans la condition.

Conditions utilisées avec les paramètres de date

Nom dans le DataBrew SDK	Nom dans la DataBrew console	Format de valeur de condition (SDK)	Description
after	Démarrer	Format de date ISO 8601 comme ou `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	La valeur du paramètre de date est postérieure à la date indiquée dans la condition.
before	Fin	Format de date ISO 8601 comme ou `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	La valeur du paramètre de date est antérieure à la date indiquée dans la condition.
relative_après	Début (relatif)	Nombre d'unités de temps positif ou négatif, comme `-48h` ou`+7d`.	La valeur du paramètre de date est postérieure à la date relative spécifiée dans la condition. Les dates relatives sont évaluées lorsque l'ensemble de données est chargé, soit lorsqu'une session interactive est initialisée, soit lorsqu'une tâche associée est démarrée. C'est le moment appelé « maintenant » dans les exemples.

Nom dans le DataBrew SDK	Nom dans la DataBrew console	Format de valeur de condition (SDK)	Description
relative_avant	Fin (relative)	Nombre d'unités de temps positif ou négatif, comme `-48h` ou`+7d`.	La valeur du paramètre de date est antérieure à la date relative spécifiée dans la condition. Les dates relatives sont évaluées lorsque l'ensembl e de données est chargé, soit lorsqu'un e session interacti ve est initialisée, soit lorsqu'une tâche associée est démarrée. C'est le moment appelé « maintenant » dans les exemples.

Si vous utilisez le SDK, fournissez des dates relatives au format suivant :`±{number_of_time_units}{time_unit}`. Vous pouvez utiliser les unités de temps suivantes :

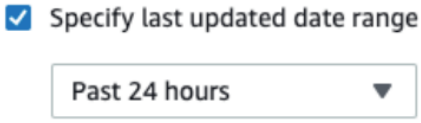
- -1h (il y a 1 heure)
- +2d (dans 2 jours)
- -120m (il y a 120 minutes)
- 5000 s (5 000 secondes dans 5 000 secondes)
- -3w (il y a 3 semaines)
- +4M (dans 4 mois)
- -1y (il y a 1 an)

Les dates relatives sont évaluées lorsque l'ensemble de données est chargé, soit lorsqu'une session interactive est initialisée, soit lorsqu'une tâche associée est démarrée. C'est le moment appelé « maintenant » dans les exemples précédents.

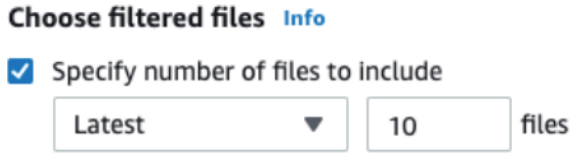
Configuration des paramètres pour les ensembles de données dynamiques

Outre le fait de fournir un chemin S3 paramétré, vous pouvez configurer d'autres paramètres pour les ensembles de données contenant plusieurs fichiers. Ces paramètres filtrent les fichiers S3 en fonction de leur date de dernière modification et limitent le nombre de fichiers.

Comme pour définir un paramètre de date dans un chemin, vous pouvez définir une plage de temps pendant laquelle les fichiers correspondants ont été mis à jour et inclure uniquement ces fichiers dans votre ensemble de données. Vous pouvez définir ces plages en utilisant soit des dates absolues telles que « 30 mars 2021 », soit des plages relatives telles que « 24 dernières heures ».



Pour limiter le nombre de fichiers correspondants, sélectionnez un nombre de fichiers supérieur à 0 et indiquez si vous voulez les fichiers correspondants les plus récents ou les plus anciens.



Types de données

Les données de chaque colonne de votre jeu de données sont converties dans l'un des types de données suivants :

- octet — nombres entiers signés sur 1 octet. La plage de nombres va de -128 à 127.
- short — nombres entiers signés sur 2 octets. La plage de nombres est comprise entre -32768 et 32767.
- entier — nombres entiers signés sur 4 octets. La plage de nombres va de -2147483648 à 2147483647.
- long — nombres entiers signés de 8 octets. La plage de nombres va de -9223372036854775808 à 9223372036854775807.

- float — nombres à virgule flottante à précision unique de 4 octets.
- double : nombres à virgule flottante à double précision de 8 octets.
- décimal : nombres décimaux signés comportant jusqu'à 38 chiffres au total et 18 chiffres après la virgule décimale.
- string — Valeurs de chaîne de caractères.
- boolean — Le type booléen a l'une des deux valeurs possibles : « vrai » et « faux » ou « oui » et « non ».
- timestamp — Valeurs comprenant les champs année, mois, jour, heure, minute et seconde.
- date — Valeurs comprenant les champs année, mois et jour.

Types de données avancés

Les types de données avancés sont des types de données DataBrew détectés dans une colonne de chaîne d'un projet et ne font donc pas partie d'un ensemble de données. Pour plus d'informations sur les types de données avancés, consultez la section Types de données avancés.

Types de données avancés

Les types de données avancés sont des types de données qui sont DataBrew détectés dans une colonne de chaîne d'un projet au moyen d'une correspondance de modèles. Lorsque vous cliquez sur une colonne de chaîne, celle-ci est marquée comme le type de données avancé correspondant si 50 % ou plus des valeurs de la colonne répondent aux critères de ce type de données.

Les types de données que DataBrew vous pouvez détecter sont les suivants :

- Date/timestamp
- SSN
- Numéro de téléphone
- E-mail
- Carte de crédit
- Gender
- Adresse IP
- URL
- Code postal

- Country
- Currency
- State
- Ville

Vous pouvez utiliser les transformations suivantes pour travailler avec des types de données avancés :

- GET_ADVANCED_DATATYPE: étant donné une colonne de chaîne, identifie le type de données avancé de la colonne, le cas échéant.
- EXTRACT_ADVANCED_DATATYPE_DETAILS: extrait les détails d'un type de données avancé.
- FILTRE_TYPE DE DONNÉES AVANCÉ: filtre une colonne source actuelle en fonction de la détection avancée des types de données.
- DRAPEAU DE TYPE DE DONNÉES AVANCÉ: crée une nouvelle colonne de drapeau basée sur les valeurs de la colonne source actuelle.

Validation de la qualité des données dans AWS Glue DataBrew

Pour garantir la qualité de vos ensembles de données, vous pouvez définir une liste de règles de qualité des données dans un ensemble de règles. Un ensemble de règles est un ensemble de règles qui comparent différentes mesures de données aux valeurs attendues. Si l'un des critères d'une règle n'est pas satisfait, l'ensemble de règles dans son ensemble échoue à la validation. Vous pouvez ensuite inspecter les résultats individuels pour chaque règle. Pour toute règle entraînant un échec de validation, vous pouvez apporter les corrections nécessaires et revalider.

Voici des exemples de règles :

- La valeur de la colonne `"APY"` est comprise entre 0 et 100
- Le nombre de valeurs manquantes dans la colonne `group_name` ne dépasse pas 5 %

Vous pouvez définir chaque règle pour une colonne individuelle ou l'appliquer indépendamment à plusieurs colonnes sélectionnées, par exemple :

- La valeur maximale ne dépasse pas 100 pour les colonnes`"rate"`,`"pay"`,`"increase"`.

Une règle peut consister en plusieurs vérifications simples. Vous pouvez définir si elles doivent toutes être vraies ou non, par exemple :

- La valeur de la colonne `"ProductId"` doit commencer par `"asin-"` ET la longueur de la valeur de la colonne `"ProductId"` est 32.

Vous pouvez vérifier les règles par rapport à des valeurs agrégées telles que `maxmin`, ou `number of duplicate values` lorsqu'une seule valeur est comparée, ou à des valeurs non agrégées dans chaque ligne d'une colonne. Dans ce dernier cas, vous pouvez également définir un seuil « satisfaisant » tel que`value in columnA > value in columnB for at least 95% of rows`.

Comme pour les informations de profil, vous pouvez définir des règles de qualité des données au niveau des colonnes uniquement pour les colonnes de types simples, telles que les chaînes et les nombres. Vous ne pouvez pas définir de règles de qualité des données pour les colonnes de

types complexes, tels que les tableaux ou les structures. Pour plus de détails sur l'utilisation des informations de profil, consultezCréation et utilisation de AWS Glue DataBrew emplois de profil.

Validation des règles de qualité des données

Une fois qu'un ensemble de règles est défini, vous pouvez l'ajouter à une tâche de profil pour validation. Vous pouvez définir plusieurs ensembles de règles pour un ensemble de données.

Par exemple, un ensemble de règles peut contenir des règles avec des critères minimalement acceptables. Un échec de validation pour cet ensemble de règles peut signifier que les données ne sont pas acceptables pour une utilisation ultérieure. Par exemple, il manque des valeurs dans les colonnes clés d'un ensemble de données utilisé pour l'apprentissage automatique. Vous pouvez utiliser un deuxième ensemble de règles avec des règles plus strictes pour vérifier si la qualité du jeu de données est telle qu'aucun nettoyage n'est nécessaire.

Vous pouvez appliquer un ou plusieurs ensembles de règles définis pour un ensemble de données donné dans une configuration de tâche de profil. Lorsque la tâche de profilage s'exécute, elle produit un rapport de validation en plus du profil de données. Le rapport de validation est disponible au même endroit que les données de votre profil. Comme pour les informations de profil, vous pouvez explorer les résultats dans la DataBrew console. Dans la vue détaillée du jeu de données, choisissez l'onglet Qualité des données pour afficher les résultats. Pour plus de détails sur l'utilisation des informations de profil, consultezCréation et utilisation de AWS Glue DataBrew emplois de profil.

Agir en fonction des résultats de validation

Lorsqu'une tâche DataBrew de profil est terminée, DataBrew envoie un CloudWatch événement Amazon avec les détails de cette tâche exécutée. Si vous avez également configuré votre tâche pour valider les règles de qualité des données, DataBrew envoie un événement pour chaque ensemble de règles validé. L'événement contient son résultat (`SUCCEEDED`,`FAILED`, ou`ERROR`) et un lien vers le rapport détaillé de validation de la qualité des données. Vous pouvez ensuite automatiser d'autres actions en invoquant l'action suivante en fonction de l'état de validation. Pour plus d'informations sur la connexion d'événements à des actions cibles, telles que les notifications Amazon SNS, les appels de AWS Lambda fonctions, etc., consultez Getting started with Amazon. EventBridge

Voici un exemple d'événement de résultat de DataBrew validation :

```
{
```

```
  "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"
  }
}
```

Vous pouvez utiliser les attributs d'événements tels que`detail-type`, `source` et les propriétés imbriquées de l'`detail`attribut, pour créer des modèles d'événements dans Amazon Eventbridge. Par exemple, un modèle d'événement correspondant à toutes les validations ayant échoué pour n'importe quelle DataBrew tâche ressemblerait à ceci :

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

Pour un exemple de création d'un ensemble de règles et de validation de ses règles, consultez. Création d'un ensemble de règles avec des règles de qualité des données Pour plus d'informations sur l'utilisation des CloudWatch événements dans DataBrew, voir Automatisation DataBrew grâce aux événements CloudWatch

Création d'un ensemble de règles avec des règles de qualité des données

Dans la procédure suivante, vous trouverez un exemple de création d'un ensemble de règles et de son application à un ensemble de données. Un ensemble de règles est un ensemble de règles qui comparent différentes mesures de données aux valeurs attendues. Vous pouvez ensuite utiliser cet ensemble de règles dans une tâche de profilage pour valider les règles de qualité des données qu'il inclut.

Pour créer un exemple d'ensemble de règles avec des règles de qualité des données

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.
2. Choisissez DQ RULES dans le volet de navigation, puis choisissez Create data quality rules eset.
3. Entrez un nom pour votre ensemble de règles. Entrez éventuellement une description de votre ensemble de règles.
4. Sous Ensemble de données associé, choisissez un ensemble de données à associer à l'ensemble de règles.

   Après avoir sélectionné un jeu de données, vous pouvez afficher le volet d'aperçu du jeu de données sur la droite.
5. Utilisez l'aperçu dans le volet d'aperçu du jeu de données pour explorer les valeurs et le schéma du jeu de données lorsque vous déterminez les règles de qualité des données à créer. L'aperçu peut vous donner un aperçu des problèmes potentiels que vous pourriez rencontrer avec les données.

   Certaines sources de données, telles que les bases de données, ne prennent pas en charge l'aperçu des données. Dans ce cas, vous pouvez exécuter une tâche de profilage sans valider au préalable les règles de qualité des données. Vous pouvez ensuite obtenir des informations sur le schéma de données et la distribution des valeurs à l'aide du profil de données.
6. Consultez l'onglet Recommandations, qui répertorie certaines suggestions de règles que vous pouvez utiliser lors de la création de votre ensemble de règles. Vous pouvez sélectionner toutes les recommandations, certaines ou aucune d'entre elles.

   Après avoir sélectionné les recommandations pertinentes, choisissez Ajouter à l'ensemble de règles.

Cela ajoutera des règles à votre ensemble de règles. Inspectez et modifiez les paramètres si nécessaire. Notez que seules les colonnes de types simples tels que les chaînes, les nombres et les booléens peuvent être utilisées dans les règles de qualité des données.

7. Choisissez Ajouter une autre règle pour ajouter une règle non couverte par les recommandations. Vous pouvez modifier le nom des règles pour faciliter l'interprétation ultérieure des résultats de validation.
8. Utilisez l'étendue du contrôle de la qualité des données pour choisir si des colonnes individuelles seront sélectionnées pour chaque vérification dans cette règle ou si elles doivent être appliquées à un groupe de colonnes que vous sélectionnez. Par exemple, si votre jeu de données comporte plusieurs colonnes numériques qui doivent avoir des valeurs comprises entre 0 et 100, vous pouvez définir la règle une seule fois et sélectionner toutes ces colonnes à vérifier par cette règle.
9. Si votre règle comporte plusieurs vérifications, dans le menu déroulant Critères de réussite des règles, indiquez si toutes les vérifications doivent être satisfaites ou lesquelles répondent aux critères.
10. Sélectionnez une vérification qui sera effectuée pour vérifier cette règle dans le menu déroulant Contrôle de qualité des données. Pour plus d'informations sur les chèques disponibles, consultezChèques disponibles.
11. Si vous avez sélectionné Contrôle individuel pour chaque colonne dans le champ du contrôle de la qualité des données, choisissez une colonne. Sélectionnez ou saisissez le nom de colonne pour cette vérification.
12. Sélectionnez les paramètres en fonction de la vérification. Certaines conditions acceptent uniquement les valeurs personnalisées fournies et d'autres acceptent également la référence à une autre colonne.
13. Si vous choisissez de vérifier les valeurs de colonne, telles que la condition Contient pour les valeurs de chaîne, vous pouvez spécifier un seuil de « dépassement ». Par exemple, si vous souhaitez qu'au moins 95 % des valeurs satisfassent à la condition, vous devez choisir Supérieur à égal comme condition de seuil, saisir 95 comme seuil et laisser « % (pourcentage) lignes » dans le menu déroulant suivant de la section Seuil. Ou si vous ne voulez pas plus de 10 lignes où la valeur est manquante, vous pouvez sélectionner Moins qu'égal comme condition, entrer 10 pour le seuil et choisir des lignes dans le menu déroulant suivant. Veuillez noter que vous pouvez obtenir des résultats différents si vous utilisez des échantillons de tailles différentes lors de la validation.
14. Ajoutez d'autres règles si nécessaire.

15. Choisissez Créer un ensemble de règles.

Création d'une tâche de profil à l'aide d'un ensemble de règles

Après avoir créé un ensemble de règles tel que décrit ci-dessus, vous êtes dirigé vers la page Règles de qualité des données, qui affiche tous les ensembles de règles de votre compte.

Pour créer un profil de travail incluant un ensemble de règles

1. Choisissez le nom de l'ensemble de règles que vous avez créé précédemment pour en afficher les détails.
2. Choisissez Créer une tâche de profil avec un ensemble de règles.

   Le nom du Job est automatiquement renseigné, mais vous pouvez le modifier si nécessaire.
3. Pour Job run sample, vous pouvez choisir d'exécuter l'ensemble de données dans son intégralité ou un nombre limité de lignes.

   Si vous choisissez d'exécuter un échantillon de taille limitée, sachez que pour certaines règles, les résultats peuvent différer de ceux de l'ensemble de données complet.
4. Pour les paramètres de sortie du job, choisissez un emplacement S3 pour le résultat du job. Choisissez n'importe quel dossier dans un compartiment Amazon S3 nommé auquel vous avez accès. Si vous entrez un nom de dossier pour ce compartiment qui n'existe pas, ce dossier est créé.

   Une fois la tâche de profilage terminée avec succès, ce dossier contiendra les profils des données et le rapport de validation des règles de qualité des données au format JSON.
5. Sous Règles de qualité des données, notez que votre ensemble de règles est répertorié sous Nom de l'ensemble de règles de qualité des données.
6. Sous Autorisations, sélectionnez ou créez un rôle pour accorder DataBrew l'accès à la lecture depuis l'emplacement d'entrée Amazon S3 et à l'emplacement de sortie de la tâche. Si aucun rôle n'est prêt, sélectionnez Créer un nouveau rôle IAM.
7. Modifiez tous les autres paramètres facultatifs comme décrit dansCréation et utilisation de AWS Glue DataBrew emplois de profil, si nécessaire.
8. Choisissez Create and run job.

Inspection des résultats de validation et mise à jour des règles de qualité des données

Une fois la tâche de création de profil terminée, vous pouvez consulter les résultats de validation de vos règles de qualité des données et, le cas échéant, mettre à jour vos règles.

Pour consulter les données de validation relatives à vos règles de qualité des données

1. Sur la DataBrew console, choisissez Afficher le profil de données. Cela permet d'afficher l'onglet Aperçu du profil de données de votre ensemble de données.
2. Choisissez l'onglet Règles de qualité des données. Dans cet onglet, vous pouvez consulter les résultats pour toutes vos règles de qualité des données.
3. Sélectionnez une règle individuelle pour plus de détails sur cette règle.

Pour toute règle dont la validation a échoué, vous pouvez apporter les corrections nécessaires.

Pour mettre à jour vos règles de qualité des données

1. Dans le volet de navigation, choisissez DQ RULES.
2. Sous Nom de l'ensemble de règles de qualité des données, choisissez le jeu de données contenant les règles que vous souhaitez modifier.
3. Choisissez la règle que vous souhaitez modifier, puis choisissez Modifier.
4. Apportez les corrections nécessaires, puis choisissez Mettre à jour l'ensemble de règles.
5. Réexécutez le job. Répétez ce processus jusqu'à ce que toutes les validations soient passées.

Chèques disponibles

Le tableau suivant répertorie les références de toutes les conditions disponibles qui peuvent être utilisées dans vos règles. Notez que les conditions agrégées ne peuvent pas être combinées avec des conditions non agrégées dans la même règle.

> **Note**
>
> Pour les utilisateurs du SDK, pour appliquer la même règle à plusieurs colonnes, utilisez l'ColumnSelectorsattribut d'une règle et spécifiez les colonnes validées en utilisant leur nom ou une expression régulière. Dans ce cas, vous devez utiliser l'implicite

CheckExpression. Par exemple, “`> :val`” pour comparer les valeurs de chacune des colonnes sélectionnées avec la valeur fournie. DataBrew utilise une syntaxe implicite pour la définition FilterExpressiondans les ensembles de données dynamiques. Si vous souhaitez spécifier une ou plusieurs colonnes pour chaque contrôle individuellement, ne définissez pas l'ColumnSelectorsattribut. Fournissez plutôt une expression explicite. Par exemple, “`:col > :val`” en tant que CheckExpressiondans une règle.

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK
Conditions du jeu de données agrégé	Nombre de lignes		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(ROWS_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
	Nombre de colonnes		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}`
	Lignes dupliquées.		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT)`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

				`< :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

Conditions statistiques des colonnes agrégées	Valeurs manquantes		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Valeurs dupliquées		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Valeurs valides		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
				
				or
				
				`"CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Des valeurs distinctes		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(DISTINCT_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "1000"}`
				
				or
				
				`"CheckExpression": "AGG(DISTINCT_VALUES_PERCENTAGE) >= :val", "SubstitutionMap": {":val", "50"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Des valeurs uniques		Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Valeurs aberrantes	Z-score seuil	Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Statistiques de distribution de valeur	Nom des statistiques (voir le tableau suivant)	Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Voir le tableau suivant pour les `STAT_NAME` valeurs possibles

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Statistiques numériques	Nom des statistiques (voir le tableau suivant)	Comparaison numérique par rapport à une valeur personnalisée	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				or
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Voir le tableau suivant pour les `STAT_NAME` valeurs possibles

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK
Non agrégé (accepte le seuil)	La valeur est exactement		Comparaison exacte par rapport à une liste de valeurs	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	La valeur n'est pas exactement		La valeur ne doit correspondre exactement à aucune valeur d'une liste	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Valeurs de chaîne		Comparaison de chaînes par rapport à une valeur personnalisée ou à une autre colonne de chaînes	`"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"}`
				
				or
				
				`"CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Valeur numériques		Comparaison numérique avec une valeur personnalisée ou une autre colonne numérique	`"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"}`
				
				or
				
				`"CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}`

Type de condition	Vérification de la qualité des données	Paramètres supplémentaires	Type de comparaison	Exemple de syntaxe du SDK

	Longueur de la chaîne de valeurs		Comparaison numérique avec une valeur personnalisée ou une autre colonne numérique	`"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"}`
				
				or
				
				`"CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}`

Comparaisons numériques

DataBrew prend en charge les opérations suivantes pour la comparaison numérique : Is equals (= =), Is not equals (! =), Inférieur à (<), Inférieur à (< =), Supérieur à (>), Supérieur à (> =) et Est compris entre (is_between:val1 et:val2).

Comparaisons de chaînes

Les comparaisons de chaînes suivantes sont prises en charge : commence par, ne commence pas par, se termine par, ne se termine pas par, contient, ne contient pas, est égal, n'est pas égal, correspond, ne correspond pas.

Le tableau suivant présente les statistiques disponibles que vous pouvez utiliser pour les statistiques de distribution de valeurs et les statistiques numériques :

Vérification de la qualité des données	Nom des statistiques	Paramètres supplémentaires	Syntaxe du SDK
Statistiques de distribution de valeur	Min		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	Max		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	Médiane		`"CheckExpression": "AGG(MEDIAN) >= :val", "Substitu`

Vérification de la qualité des données	Nom des statistiques	Paramètres supplémentaires	Syntaxe du SDK
			`tionMap": {":val", "50"}`
	Mean		`"CheckExpression": "AGG(MEAN) <= :val", "SubstitutionMap": {":val", "10"}`
	Mode		`"CheckExpression": "AGG(MODE) > :val", "SubstitutionMap": {":val", "0"}`
	Écart-type standard		`"CheckExpression": "AGG(STANDARD_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`

Vérification de la qualité des données	Nom des statistiques	Paramètres supplémentaires	Syntaxe du SDK
	Entropie		`"CheckExpression": "AGG(ENTROPY) > :val", "SubstitutionMap": {":val", "0"}`
Statistiques numériques	Somme		`"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}`
	Kurtosis		`"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}`
	Asychité		`"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}`

Vérification de la qualité des données	Nom des statistiques	Paramètres supplémentaires	Syntaxe du SDK
	Variance		`"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}`
	Déviation absolue		`"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Quantile	Quantile : l'un des nombres suivants : « 0,25 », « 0,5 », « 0,75 »	`"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}`

Création et utilisation AWS Glue DataBrew projects

En AWS Glue DataBrew, un projet est la pièce maîtresse de vos efforts d'analyse et de transformation des données.

Lorsque vous créez un projet, vous réunissez deux éléments fondamentaux :

- Un ensemble de données, pour fournir un accès en lecture seule à vos données sources. Pour de plus amples informations, veuillez consulter Connexion aux données avec AWS Glue DataBrew.
- Une recette pour appliquer des transformations de DataBrew données à l'ensemble de données. Pour de plus amples informations, veuillez consulter Création et utilisation AWS Glue DataBrew recipes.

La DataBrew console présente votre projet dans une interface utilisateur intuitive et hautement interactive. Il vous encourage à expérimenter des centaines de transformations de données, afin que vous puissiez découvrir comment elles fonctionnent et quel effet elles ont sur vos données.

Les données que vous voyez dans la vue du projet sont un échantillon de votre ensemble de données. Les ensembles de données pouvant être très volumineux, avec des milliers, voire des millions de lignes, l'utilisation d'un échantillon permet de garantir que la DataBrew console reste réactive lorsque vous transformez les exemples de données de différentes manières. Par défaut, l'échantillon comprend les 500 premières lignes de données de l'ensemble de données. Vous pouvez choisir différents paramètres pour la taille de l'échantillon et les lignes choisies.

Au fur et à mesure que vous transformez les exemples de données, DataBrew cela vous aide à créer et à affiner la recette du projet, à savoir une série étape par étape des transformations que vous avez appliquées jusqu'à présent. Votre recette en cours est automatiquement enregistrée, ce qui vous permet de quitter la vue du projet à tout moment, d'y revenir plus tard et de reprendre là où vous vous êtes arrêté.

Lorsque votre recette est prête à être utilisée, vous pouvez la publier. La publication d'une recette la met à la disposition du sous-système des DataBrew tâches, où vous pouvez appliquer la recette à l'ensemble de votre jeu de données ou créer un profil de données complet qui vous permet de comprendre la structure, le contenu et les caractéristiques statistiques de vos données.

Rubriques

- Vue d'ensemble d'une session de DataBrew projet
- Suppression d’un projet

Création d’un projet

Pour créer un projet, procédez comme suit.

Pour créer un projet

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la.
2. Dans le volet de navigation, sélectionnez PROJECTS. Choisissez ensuite Créer un projet.
3. Tapez un nom pour votre projet. Choisissez ensuite une recette à joindre à votre projet :
   - Choisissez Créer une nouvelle recette si vous recommencez depuis le début. Cela crée une nouvelle recette vide et l'associe à votre projet.
   - Choisissez Modifier la recette existante si vous avez déjà publié une recette que vous souhaitez utiliser pour ce projet. Si la recette est actuellement attachée à un autre projet ou si des tâches sont définies pour celle-ci, vous ne pouvez pas l'utiliser dans votre nouveau projet. Choisissez Parcourir les recettes pour voir quelles recettes sont disponibles.
   - Choisissez Importer les étapes depuis la recette si vous avez une recette existante qui a déjà été publiée et que vous souhaitez importer ses étapes, puis procédez comme suit :
     1. Choisissez Parcourir les recettes pour voir quelles recettes sont disponibles.
     2. Choisissez la version publiée de la recette que vous souhaitez utiliser. Une recette peut avoir plusieurs versions, selon la fréquence à laquelle vous l'avez publiée en mode projet.
     3. Choisissez Afficher les étapes de la recette pour examiner les transformations de données contenues dans la recette.
4. Une fois que vous avez une recette, choisissez le jeu de données avec lequel vous souhaitez travailler dans le volet Sélectionner un jeu de données :
   - Mes ensembles de données — Choisissez un jeu de données que vous avez créé précédemment. Pour plus d'informations, consultez Création d’un projet.
   - Fichiers d'exemple — Créez un nouvel ensemble de données basé sur des exemples de données gérés par AWS. Ces exemples de données constituent un excellent moyen d'explorer ce que DataBrew vous pouvez faire, sans avoir à fournir vos propres données. Assurez-vous de saisir un nom pour votre jeu de données.

- Nouvel ensemble de données — Créez un nouveau jeu de données. Pour de plus amples informations, veuillez consulter Création d'un projet.

5. Pour les autorisations d'accès, choisissez un rôle Gestion des identités et des accès AWS(IAM) qui permet de DataBrew lire depuis votre emplacement d'entrée Amazon S3. Pour un emplacement S3 appartenant à votre AWS compte, vous pouvez choisir le rôle `AwsGlueDataBrewDataAccessRole` géré par le service. Cela permet d'accéder DataBrew aux ressources S3 que vous possédez.
6. Dans le volet Échantillonnage, vous trouverez des options permettant DataBrew de créer un échantillon de données à partir de votre ensemble de données.

   Dans Type, choisissez le mode d' DataBrew obtention des lignes de votre jeu de données :

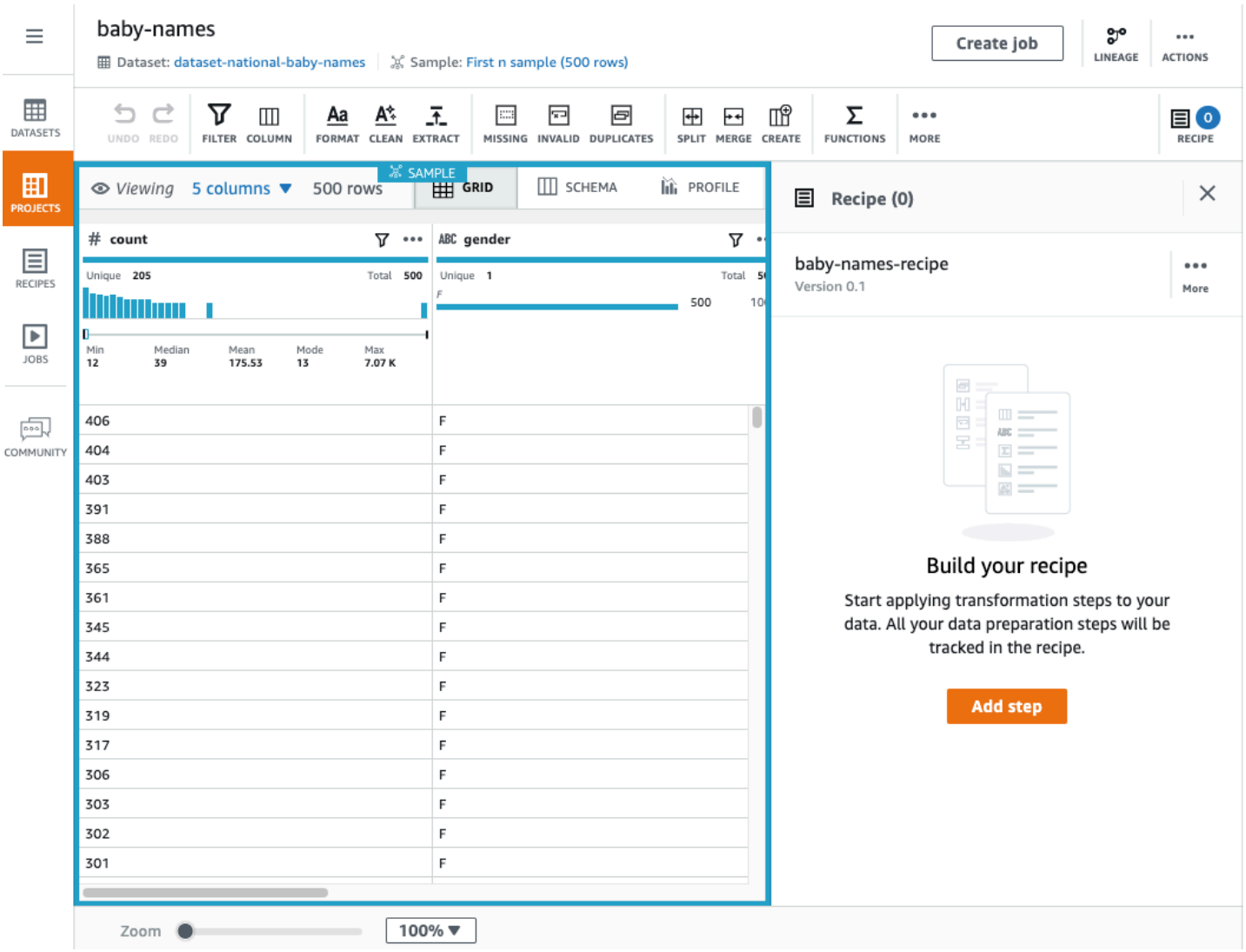
   - Utilisez les n premières lignes pour créer un échantillon basé sur les premières lignes du jeu de données.
   - Utilisez des lignes aléatoires pour créer un échantillon basé sur une sélection aléatoire de lignes du jeu de données.
   - Choisissez le nombre de lignes à afficher dans l'échantillon : 500, 1 000, 2 500, ou une taille d'échantillon personnalisée, jusqu'à un maximum de 5 000 lignes. Un échantillon plus petit permet d' DataBrew effectuer des transformations plus rapidement, ce qui vous permet de gagner du temps lors de l'élaboration de votre recette. Un échantillon plus grand reflète plus précisément la composition des données sources sous-jacentes. Cependant, l'initialisation des sessions de projet et les transformations interactives sont plus lentes.
7. (Facultatif) Choisissez Tags pour associer des balises à votre jeu de données.

   Les balises sont de simples étiquettes composées d'une clé définie par l'utilisateur et d'une valeur facultative qui peuvent faciliter la gestion, la recherche et le filtrage DataBrew des projets par objectif, propriétaire, environnement ou autres critères.
8. Lorsque les paramètres sont tels que vous le souhaitez, choisissez Create job.

DataBrew crée un nouvel ensemble de données si nécessaire, crée une nouvelle recette si nécessaire, crée l'échantillon de données et crée une session de projet interactive. Ce processus peut prendre quelques minutes. Lorsque le projet est prêt à être utilisé, vous pouvez commencer à travailler avec l'échantillon de données.

Vue d'ensemble d'une session de DataBrew projet

Dans une session de DataBrew projet, vous travaillez dans un espace de travail interactif.



Le volet de gauche affiche la vue actuelle de vos données. Le volet droit affiche la recette de transformation du projet, qui est actuellement vide.

Dans le coin supérieur droit de la grille de données, vous trouverez trois onglets : `GRID`SCHEMA, et. `PROFILE` Le choix de l'un de ces onglets permet d'afficher une vue correspondante dans l'espace de travail ; ces vues sont décrites ci-dessous.

Vue de la grille

La vue en grille est la vue par défaut, où l'échantillon est présenté sous forme de tableau. Utilisez la procédure suivante pour une brève présentation de la vue en grille.

Pour faire une présentation de la vue en grille

1. Commencez par visualiser l'ensemble de l'espace :

   a. Faites défiler l'écran vers la gauche et vers la droite pour voir toutes les colonnes.

   b. Faites défiler l'écran vers le haut ou vers le bas pour voir toutes les valeurs des données.

   c. Utilisez la commande de zoom en bas de l'espace de travail pour régler le niveau d'agrandissement de la grille.

2. En haut à droite, visualisez le nombre de colonnes de l'échantillon affichées et le nombre actuel de lignes de l'échantillon.

   Pour modifier les colonnes affichées, cliquez sur le lien N colonnes (où N est le nombre de colonnes actuellement affichées). Choisissez les colonnes souhaitées, puis choisissez Afficher les colonnes sélectionnées.

3. Vous pouvez maintenant commencer à expérimenter des DataBrew transformations. Essayez les éléments suivants :

   a. Dans la barre d'outils de transformation, choisissez Choisir le format, Passer en majuscules.

   b. Pour Colonne source, choisissez une colonne contenant des données de caractère.

   c. Conservez les valeurs par défaut des autres paramètres.

   d. Pour voir à quoi ressembleront les données transformées, choisissez Aperçu des modifications. Ensuite, pour ajouter cette transformation à votre recette, choisissez Appliquer.

   Chaque fois que vous appliquez une transformation de données, DataBrew ajoutez-la à la copie de travail de votre recette. Cela apparaît sur le côté droit de votre espace de travail.

4. Essayez les éléments suivants :

   a. Dans la barre d'outils de transformation, choisissez Créer en fonction d'une fonction.

   b. Pour Sélectionner une fonction, choisissez`SQUARE ROOT`.

   c. Pour Colonne source, choisissez une colonne contenant des données numériques.

   d. Conservez les autres paramètres par défaut,.

   e. Choisissez Prévisualiser les modifications pour voir à quoi ressemblent les données transformées. Ensuite, pour ajouter cette transformation à votre recette, choisissez Appliquer.

5. Réduisez le volet des recettes en haut à droite en choisissant RECETTE. Pour développer le volet des recettes, sélectionnez à nouveau RECETTE.

Publier une nouvelle version de votre recette

Au fur et à mesure que vous appliquez des transformations, le nombre d'étapes de la recette augmente. À tout moment, vous pouvez publier une nouvelle version de votre recette. La publication d'une recette la rend disponible ailleurs dans DataBrew. Vous pouvez ainsi exécuter une tâche de recette pour transformer l'ensemble de données de votre ensemble de données, au lieu de transformer uniquement l'échantillon de données du projet.

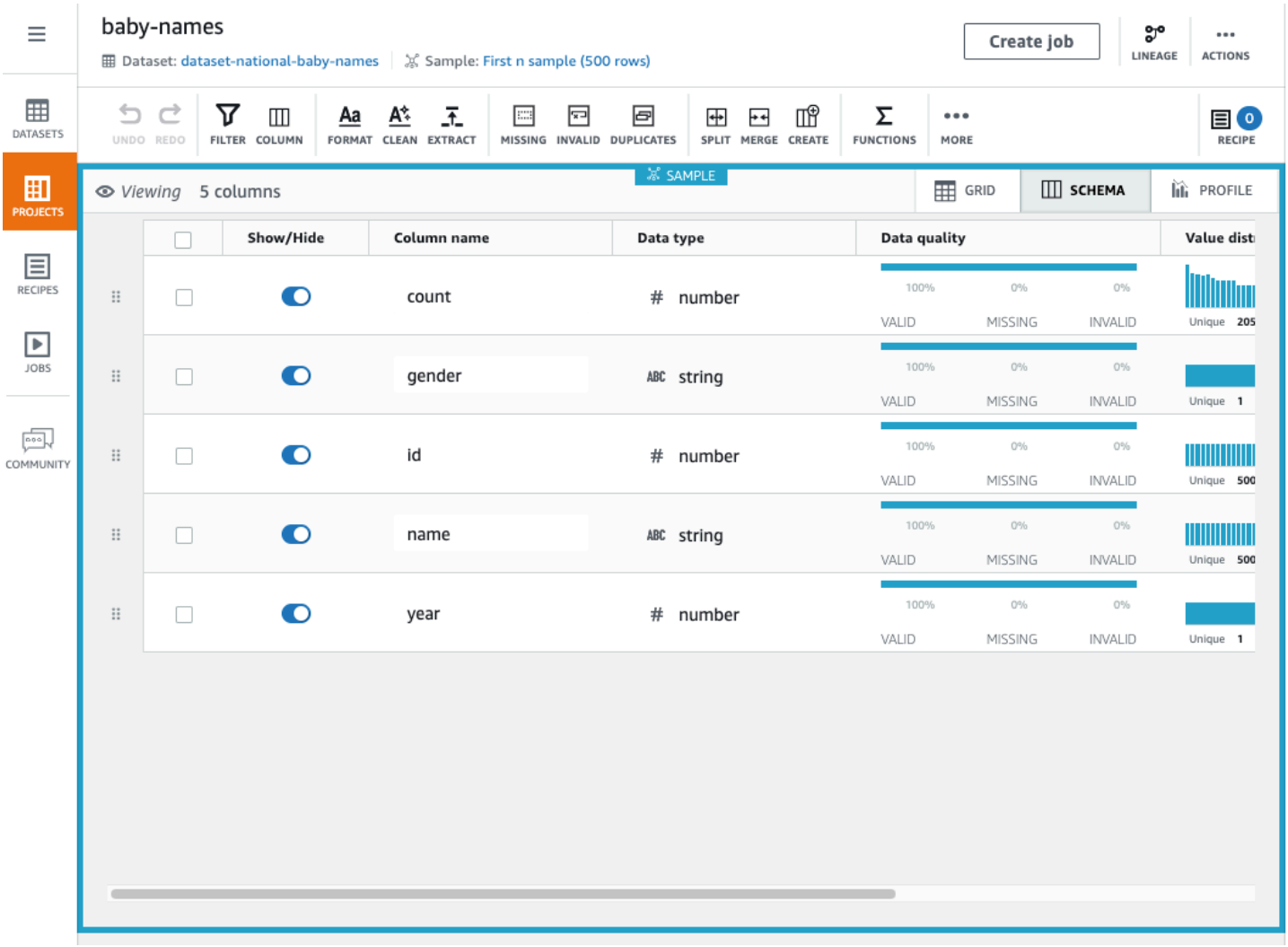
La publication de recettes encourage également une approche progressive et itérative du développement de recettes : vous pouvez publier de nouvelles versions de votre recette au fur et à mesure, afin de pouvoir revenir à la « dernière bonne version connue » si nécessaire.

Pour publier une nouvelle version d'une recette

- Dans le volet des recettes, choisissez Publier. Entrez une description pour cette version de la recette, puis choisissez Publier.

Vue du schéma

Si vous choisissez l'onglet SCHÉMA, la vue change, comme indiqué dans la capture d'écran ci-dessous.



En mode schéma, vous pouvez consulter les statistiques relatives aux valeurs de données de chaque colonne.

Dans la colonne située à l'extrême gauche, à côté de Show/Hide, sélectionnez l'une des colonnes de données. Le volet Détails de la colonne apparaît à droite. Ce volet affiche un résumé des statistiques relatives aux valeurs des colonnes.

Vous pouvez renommer une colonne en saisissant un nouveau nom pour Nom de colonne.

Vous pouvez réorganiser l'ordre des colonnes en faisant glisser les colonnes.

Vue du profil

Si vous choisissez l'onglet PROFIL, vous pouvez consulter des informations volumétriques détaillées sur votre projet. Avant cela, vous devez exécuter une DataBrew tâche pour créer le profil.

Pour passer en revue l'affichage du profil

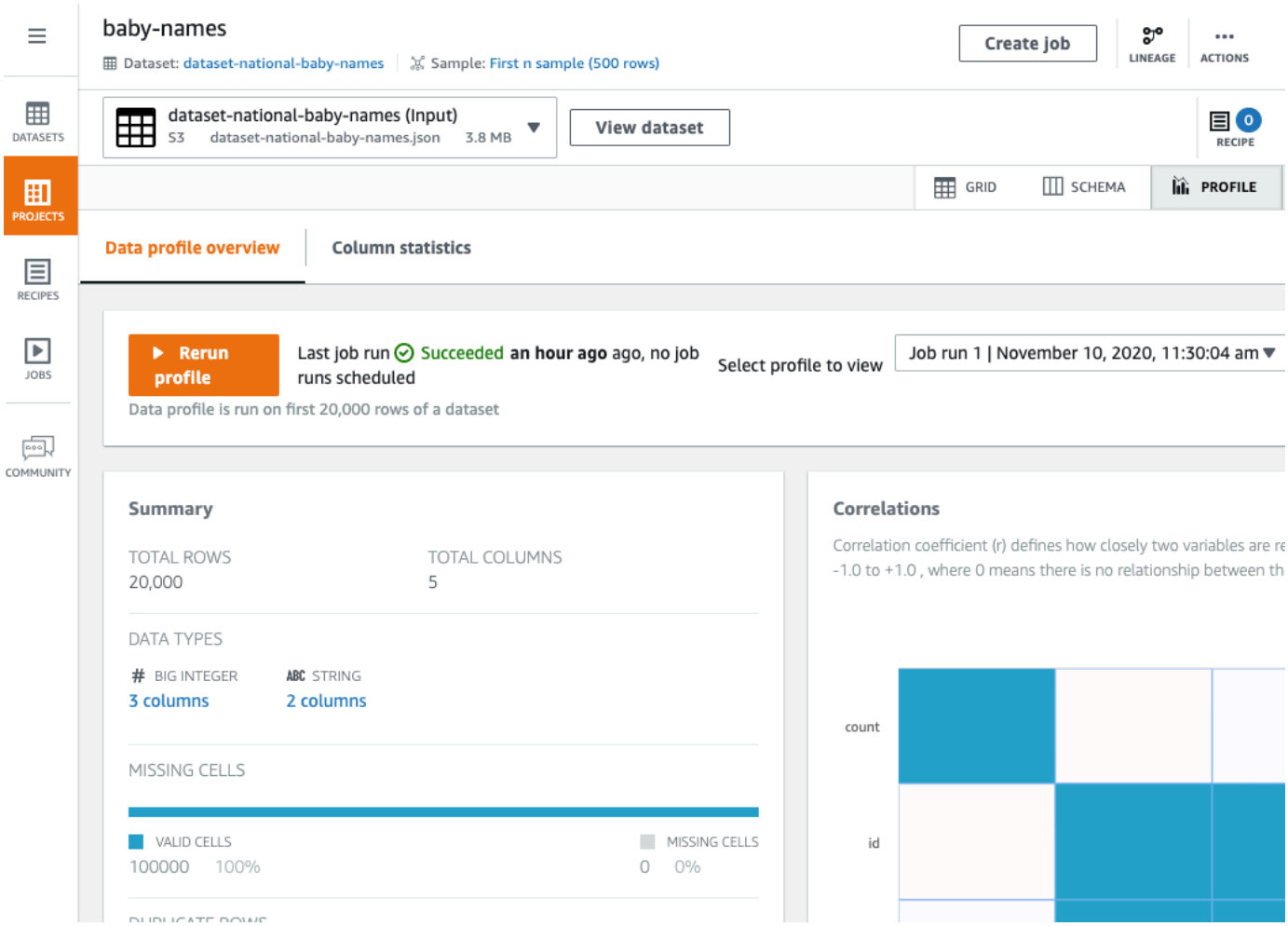
1. Choisissez Créer une tâche, puis entrez le nom de votre tâche.
2. Pour Job output, choisissez CSV comme type de fichier.
3. Recherchez ou créez un compartiment et un dossier Amazon S3 dans votre AWS compte dans lesquels vous souhaitez enregistrer le résultat DataBrew de la tâche :

   - Si vous possédez déjà ce compartiment et ce dossier Amazon S3, choisissez Browse et localisez-les. Assurez-vous que vous disposez des autorisations d'écriture pour les deux.
   - Si vous n'avez pas ce compartiment et ce dossier Amazon S3, créez-les :

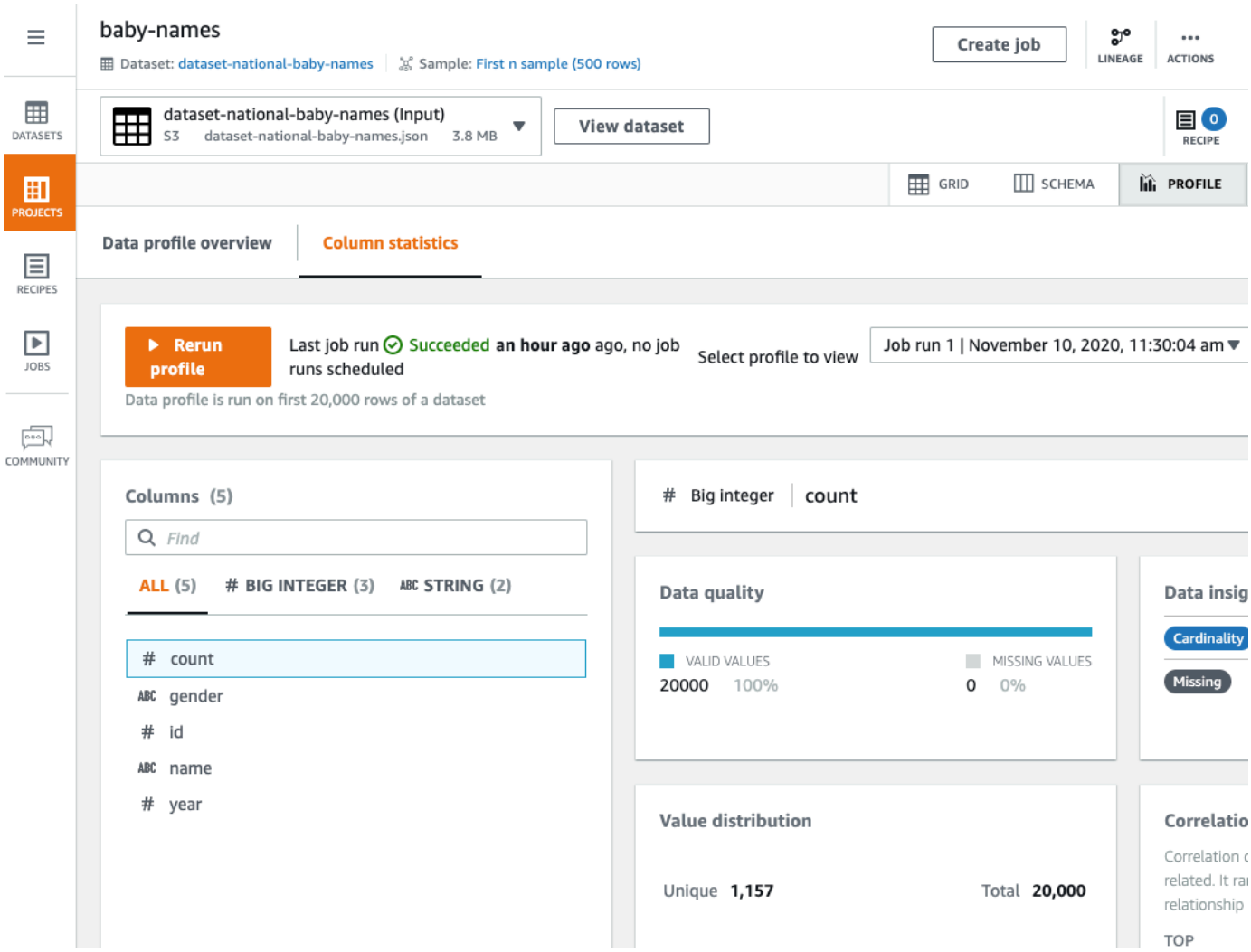
     1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
     2. Si vous n'avez pas de compartiment Amazon S3, choisissez Create bucket. Dans le champ Nom du compartiment, entrez un nom unique pour votre nouveau compartiment. Choisissez Créer un compartiment.
     3. Dans la liste des buckets, choisissez celui que vous souhaitez utiliser.
     4. Choisissez Créer un dossier. Dans Nom du dossier`databrew-output`, entrez et choisissez Créer un dossier.
4. Pour les autorisations d'accès, choisissez un rôle IAM qui permet DataBrew d'écrire sur votre emplacement de sortie Amazon S3.

   Pour un emplacement S3 appartenant à votre AWS compte, vous pouvez choisir le rôle `AwsGlueDataBrewDataAccessRole` géré par le service. Cela permet d'accéder DataBrew aux ressources S3 que vous possédez.
5. Conservez les autres paramètres par défaut, puis choisissez Create and run job.
6. Une fois le travail terminé, l'espace de travail affiche un résumé graphique du profil de données.

   L'onglet Aperçu du profil de données présente un résumé détaillé des caractéristiques de vos données, comme le montre la capture d'écran ci-dessous.



L'onglet Statistiques des colonnes affiche une ventilation colonne par colonne des valeurs de données :



Suppression d'un projet

Si vous n'avez plus besoin d'un projet, vous pouvez le supprimer.

Pour supprimer un projet

1. Dans le volet de navigation, sélectionnez PROJECTS.
2. Choisissez le projet que vous souhaitez supprimer, puis dans Actions, sélectionnez Supprimer. .

Création et utilisation AWS Glue DataBrew recipes

Dans DataBrew, une recette est un ensemble d'étapes de transformation de données. Vous pouvez appliquer ces étapes à un échantillon de vos données ou appliquer la même recette à un ensemble de données.

Le moyen le plus simple de développer une recette consiste à créer un DataBrew projet dans lequel vous pouvez travailler de manière interactive avec un échantillon de vos données. Pour plus d'informations, consultez. Création et utilisation AWS Glue DataBrew projects Dans le cadre du processus de création de projet, une nouvelle recette (vide) est créée et jointe au projet. Vous pouvez ensuite commencer à élaborer votre recette en ajoutant des transformations de données.

> **Note**
>
> Vous pouvez inclure jusqu'à 100 transformations de données dans une seule DataBrew recette.

Au fur et à mesure que vous développez votre recette, vous pouvez enregistrer votre travail en publiant la recette. DataBrew tient à jour une liste des versions publiées de votre recette. Vous pouvez utiliser n'importe quelle version publiée dans une tâche de recette pour exécuter la recette (dans une tâche de recette) afin de transformer votre ensemble de données. Vous pouvez également télécharger une copie des étapes de la recette afin de pouvoir réutiliser la recette dans d'autres projets ou dans d'autres transformations de jeux de données.

Vous pouvez également développer des DataBrew recettes par programmation, en utilisant le AWS Command Line Interface(AWS CLI) ou l'un des AWS SDK. Dans l' DataBrew API, les transformations sont appelées actions de recette.

> **Note**
>
> Dans une session de DataBrew projet interactive, chaque transformation de données que vous appliquez entraîne un appel à l' DataBrew API. Ces appels d'API se produisent automatiquement, sans que vous ayez à connaître les détails des coulisses.

Même si vous n'êtes pas programmeur, il est utile de comprendre la structure d'une recette et la façon dont les actions DataBrew de la recette sont organisées.

Rubriques

Publication d'une nouvelle version de recette

Vous publiez de nouvelles versions d'une recette dans une session de DataBrew projet interactive.

Pour publier une nouvelle version de recette

1. Dans le volet des recettes, choisissez Publier.
2. Entrez une description pour cette version de la recette, puis choisissez Publier.

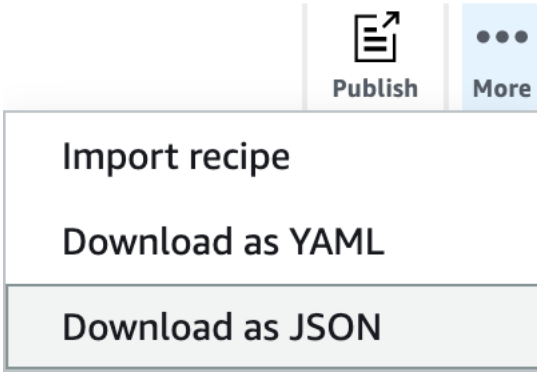
Vous pouvez consulter toutes vos recettes publiées, ainsi que leurs versions, en choisissant PROJETS dans le volet de navigation.

Définition de la structure d'une recette

Lorsque vous créez un projet pour la première fois à l'aide de la DataBrew console, vous définissez une recette à associer à ce projet. Si vous n'avez pas de recette existante, la console en crée une pour vous.

Lorsque vous travaillez sur votre projet dans la console, vous utilisez la barre d'outils de transformation pour appliquer des actions aux exemples de données de votre ensemble de données. La console affiche les étapes de la recette, ainsi que l'ordre de ces étapes, au fur et à mesure que vous poursuivez la création de la recette. Vous pouvez répéter et affiner la recette jusqu'à ce que vous soyez satisfait des étapes.

DansDémarrage avec AWS Glue DataBrew, vous créez une recette pour transformer un ensemble de données de jeux d'échecs célèbres. Vous pouvez télécharger une copie des étapes de la recette en choisissant Télécharger au format JSON ou Télécharger au format YAML, comme indiqué dans la capture d'écran ci-dessous.



Le fichier JSON téléchargé contient les actions de recette correspondant aux transformations que vous avez ajoutées à votre recette.

Une nouvelle recette ne comporte aucune étape. Vous pouvez représenter une nouvelle recette sous la forme d'une liste JSON vide, comme indiqué ci-dessous.

```
[ ]
```

Voici un exemple d'un tel fichier, pour`chess-project-recipe`. La liste JSON contient plusieurs objets qui décrivent les étapes de la recette. Chaque objet de la liste JSON est entouré d'accolades (){ }. Les lignes JSON sont délimitées par des virgules.

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

Il est plus facile de voir que chaque action est une ligne individuelle si nous ajoutons uniquement de nouvelles lignes pour les nouvelles actions, comme indiqué ci-dessous.

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

Les actions sont effectuées de manière séquentielle, dans le même ordre que dans le fichier :

- `REMOVE_VALUES`— Pour filtrer toutes les parties dont la note d'un joueur est inférieure à 1 800, la note minimale requise pour être un joueur d'échecs de classe A. Cette action se produit à deux reprises : l'une pour éliminer les joueurs du côté noir qui ne sont pas au moins des joueurs de classe A, et l'autre pour supprimer les joueurs du côté blanc qui n'ont pas atteint ce niveau.
- `GROUP_BY`— Pour résumer les données. Dans ce cas, GROUP_BY trie les lignes en groupes en fonction des valeurs de `winner` (`blacketwhite`). Chacun de ces groupes est ensuite décomposé davantage, triant les lignes en sous-groupes en fonction des valeurs de `victory_status` (`mate`, `resignoutoftime`, et`draw`). Enfin, le nombre d'occurrences pour chaque sous-groupe est compté. Le résumé obtenu remplace ensuite l'échantillon de données d'origine.
- `REMOVE_VALUES`— Pour supprimer les résultats des parties terminées par`draw`.
- `REPLACE_TEXT`— Pour modifier les valeurs de`victory_status`. Il existe trois occurrences de cette action, une pour `materesign`, et une pour chacune. `oufoftime`

Dans une session de DataBrew projet interactive, chacune `RecipeAction` correspond à une transformation de données que vous appliquez à un échantillon de données.

DataBrew fournit plus de 200 actions de recette. Pour de plus amples informations, veuillez consulter Étape de recette et référence des fonctions.

Utilisation de conditions

Vous pouvez utiliser des conditions pour réduire la portée d'une action de recette. Les conditions sont utilisées dans les transformations qui filtrent les données, par exemple pour supprimer les lignes indésirables en fonction d'une valeur de colonne particulière.

Examinons de plus près les actions d'une recette à partir de`chess-project-recipe`.

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

Cette transformation lit les valeurs de la `black_rating` colonne. La `ConditionExpressions` liste détermine les critères de filtrage : toute ligne dont `black_rating` la valeur est inférieure à 1 800 est supprimée du jeu de données.

Une transformation ultérieure de la recette fait la même chose, car`white_rating`. De cette façon, les données sont limitées aux jeux où chaque joueur (noir ou blanc) est classé dans la classe A ou plus.

Voici un autre exemple de condition appliquée à une colonne de données de caractères.

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

Cette transformation lit les valeurs de la `winner` colonne, recherche la valeur `draw` et supprime ces lignes. De cette façon, les données sont limitées aux seuls jeux où il y a eu un gagnant clair.

DataBrew prend en charge les conditions suivantes :

- `IS`— La valeur de la colonne est identique à la valeur fournie dans la condition.
- `IS_NOT`— La valeur de la colonne n'est pas la même que celle fournie dans la condition.
- `IS_BETWEEN`— La valeur de la colonne se situe entre les `LESS_THAN_EQUAL` paramètres `GREATER_THAN_EQUAL` et.
- `CONTAINS`— La valeur de chaîne de la colonne contient la valeur fournie dans la condition.
- `NOT_CONTAINS`— La valeur de la colonne ne contient pas la chaîne de caractères fournie dans la condition.
- `STARTS_WITH`— La valeur de la colonne commence par la chaîne de caractères fournie dans la condition.
- `NOT_STARTS_WITH`— La valeur de la colonne ne commence pas par la chaîne de caractères fournie dans la condition.
- `ENDS_WITH`— La valeur de la colonne se termine par la chaîne de caractères fournie dans la condition.
- `NOT_ENDS_WITH`— La valeur de la colonne ne se termine pas par la chaîne de caractères fournie dans la condition.
- `LESS_THAN`— La valeur de la colonne est inférieure à la valeur spécifiée dans la condition.
- `LESS_THAN_EQUAL`— La valeur de la colonne est inférieure ou égale à la valeur fournie dans la condition.
- `GREATER_THAN`— La valeur de la colonne est supérieure à la valeur fournie dans la condition.
- `GREATER_THAN_EQUAL`— La valeur de la colonne est supérieure ou égale à la valeur fournie dans la condition.
- `IS_INVALID`— Le type de données de la valeur de la colonne est incorrect.
- `IS_MISSING`— Il n'y a aucune valeur dans la colonne.

Création, exécution et planification AWS Glue DataBrew jobs

AWS Glue DataBrew possède un sous-système de tâches qui répond à deux objectifs :

1. Appliquer une recette de transformation de données à un DataBrew ensemble de données. Vous le faites avec un travail de DataBrew recette.
2. Analyse d'un ensemble de données pour créer un profil complet des données. Vous le faites dans le cadre d'un emploi DataBrew profilé.

Rubriques

Création et utilisation de AWS Glue DataBrew jobs de recettes

Utilisez une tâche de DataBrew recette pour nettoyer et normaliser les données d'un DataBrew ensemble de données et écrivez le résultat dans un emplacement de sortie de votre choix. L'exécution d'une tâche de recette n'affecte pas l'ensemble de données ni les données source sous-jacentes. Lorsqu'une tâche est exécutée, elle se connecte aux données source en lecture seule. La sortie de la tâche est écrite dans un emplacement de sortie que vous définissez dans Amazon S3, dans ou dans une base AWS Glue Data Catalog de données JDBC prise en charge.

Utilisez la procédure suivante pour créer une tâche de DataBrew recette.

Pour créer une tâche de recette

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.
2. Choisissez JOBS dans le volet de navigation, choisissez l'onglet Recipe jobs, puis choisissez Create job.
3. Entrez un nom pour votre tâche, puis choisissez Créer une tâche de recette.
4. Pour la saisie Job, entrez les détails de la tâche que vous souhaitez créer : le nom du jeu de données à traiter et la recette à utiliser.

   Une tâche de recette utilise une DataBrew recette pour transformer un ensemble de données. Pour utiliser une recette, assurez-vous de la publier au préalable.

5. Configurez les paramètres de sortie de votre tâche.

   Indiquez une destination pour le résultat de votre travail. Si aucune DataBrew connexion n'est configurée pour votre destination de sortie, configurez-la d'abord dans l'onglet DATASETS comme décrit dansConnexions prises en charge pour les sources de données et les sorties. Choisissez l'une des destinations de sortie suivantes :

   - Amazon S3, avec ou sans AWS Glue Data Catalog support
   - Amazon Redshift, avec ou sans assistance AWS Glue Data Catalog
   - JDBC
   - Tables Snowflake
   - Tables de base de données Amazon RDS avec AWS Glue Data Catalog support. Les tables de base de données Amazon RDS prennent en charge les moteurs de base de données suivants :
     - Amazon Aurora
     - MySQL
     - Oracle
     - PostgreSQL
     - Microsoft SQL Server
   - Amazon S3 avec AWS Glue Data Catalog support.

   Pour la AWS Glue Data Catalog sortie basée sur AWS Lake Formation, ne DataBrew prend en charge que le remplacement de fichiers existants. Dans cette approche, les fichiers sont remplacés afin de conserver intactes vos autorisations Lake Formation existantes pour votre rôle d'accès aux données. DataBrew Donne également la priorité à l'emplacement Amazon S3 indiqué dans le AWS Glue Data Catalog tableau. Ainsi, vous ne pouvez pas modifier l'emplacement Amazon S3 lors de la création d'une tâche de recette.

   Dans certains cas, l'emplacement Amazon S3 indiqué dans le résultat de la tâche est différent de celui indiqué dans le tableau du catalogue de données. Dans ces cas, DataBrew met automatiquement à jour la définition de tâche avec l'emplacement Amazon S3 indiqué dans la table du catalogue. Il le fait lorsque vous mettez à jour ou démarrez vos tâches existantes.

6. Pour les destinations de sortie Amazon S3 uniquement, d'autres choix s'offrent à vous :

a. Choisissez l'un des formats de sortie de données disponibles pour Amazon S3, la compression facultative et un séparateur personnalisé en option. Les délimiteurs pris en charge pour les fichiers de sortie sont les mêmes que pour les fichiers d'entrée : virgule, deux-points, point-virgule, tube, tabulation, curseur, barre oblique inversée et espace. Pour plus de détails sur le formatage, consultez le tableau suivant.

Format	Extension de fichier (non compressée)	Extensions de fichiers (compressées)
Comma-separated valeurs	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated valeurs	`.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	Non pris en charge	`.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`

Format	Extension de fichier (non compressée)	Extensions de fichiers (compressées)
xml	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`
JSON (format de lignes JSON uniquement)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tableau Hyper	Non pris en charge	Non applicable

b.

Choisissez de sortir un seul fichier ou plusieurs fichiers. Il existe trois options pour la sortie de fichiers avec Amazon S3 :

- Génération automatique de fichiers (recommandé) : a DataBrew déterminé le nombre optimal de fichiers de sortie.
- Sortie de fichier unique — Permet de générer un seul fichier de sortie. Cette option peut augmenter le temps d'exécution des tâches car un post-traitement est nécessaire.
- Sortie de fichiers multiples — Avez-vous spécifié le nombre de fichiers pour la sortie de votre tâche ? Les valeurs valides sont 2 à 999. Le nombre de fichiers générés peut être inférieur à celui que vous spécifiez si le partitionnement des colonnes est utilisé ou si le nombre de lignes de la sortie est inférieur au nombre de fichiers que vous spécifiez.

c.

(Facultatif) Choisissez le partitionnement des colonnes pour la sortie des tâches de recette.

Le partitionnement en colonnes constitue un autre moyen de partitionner le résultat de votre travail de recette en plusieurs fichiers. Le partitionnement des colonnes peut être utilisé avec une sortie Amazon S3 nouvelle ou existante ou avec une nouvelle sortie du catalogue de données Amazon S3. Il ne peut pas être utilisé avec les tables Amazon S3 du catalogue de données existantes. Les fichiers de sortie sont basés sur les valeurs des noms de colonnes que vous spécifiez. Si les noms de colonnes que vous spécifiez sont uniques, les chemins de dossier Amazon S3 obtenus sont basés sur l'ordre des noms de colonnes.

Pour un exemple de partitionnement de colonnesExemple de partitionnement de colonnes, reportez-vous à la section suivante.

7. (Facultatif) Choisissez Activer le chiffrement pour la sortie de la tâche pour chiffrer la sortie de la tâche qui DataBrew écrit sur votre emplacement de sortie, puis choisissez la méthode de chiffrement :

   - Utiliser le SSE-S3 chiffrement — La sortie est chiffrée à l'aide d'un chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3.
   - Use AWS Key Management Service(AWS KMS) — La sortie est cryptée à l'aide de AWS KMS. Pour utiliser cette option, choisissez le Amazon Resource Name (ARN) de la AWS KMS clé que vous souhaitez utiliser. Si vous n'avez pas de AWS KMS clé, vous pouvez en créer une en choisissant Créer une AWS KMS clé.

8. Pour les autorisations d'accès, choisissez un rôle Gestion des identités et des accès AWS(IAM) qui permet d' DataBrew écrire sur votre emplacement de sortie. Pour un établissement appartenant à votre AWS compte, vous pouvez choisir le rôle `AwsGlueDataBrewDataAccessRole` géré par le service. Cela permet d'accéder DataBrew aux AWS ressources que vous possédez.

9. Dans le volet Paramètres avancés des tâches, vous pouvez choisir d'autres options concernant le mode d'exécution de votre tâche :

   - Nombre maximum d'unités : DataBrew traite les tâches à l'aide de plusieurs nœuds de calcul et s'exécute en parallèle. Le nombre de nœuds par défaut est de 5. Le nombre maximum de nœuds est de 149.
   - Délai d'expiration d'une tâche : si une tâche prend plus de minutes que le nombre de minutes que vous avez défini ici pour s'exécuter, elle échoue avec une erreur de temporisation. La valeur par défaut est de 2 880 minutes, soit 48 heures.
   - Nombre de tentatives : si une tâche échoue en cours d'exécution, DataBrew vous pouvez essayer de l'exécuter à nouveau. Par défaut, la tâche n'est pas réessayée.
   - Activer Amazon CloudWatch Logs pour le travail : permet DataBrew de publier des informations de diagnostic dans CloudWatch Logs. Ces journaux peuvent être utiles à des fins de résolution des problèmes ou pour obtenir plus de détails sur le traitement de la tâche.

10. Pour les tâches planifiées, vous pouvez appliquer un calendrier de DataBrew travail afin que votre tâche soit exécutée à un moment précis ou de manière récurrente. Pour de plus amples informations, veuillez consulter Automatiser l'exécution des tâches selon un calendrier.

11. Lorsque les paramètres sont tels que vous le souhaitez, choisissez Create job. Ou, si vous souhaitez exécuter le travail immédiatement, choisissez Create and run job.

Vous pouvez suivre la progression de votre tâche en vérifiant son statut pendant son exécution. Lorsque l'exécution de la tâche est terminée, le statut passe à Successed. La sortie de la tâche est désormais disponible à l'emplacement de sortie que vous avez choisi.

DataBrew enregistre la définition de votre tâche afin que vous puissiez exécuter la même tâche ultérieurement. Pour réexécuter une tâche, choisissez Jobs dans le volet de navigation. Choisissez le travail avec lequel vous souhaitez travailler, puis sélectionnez Exécuter le travail.

Exemple de partitionnement de colonnes

À titre d'exemple de partitionnement de colonnes, supposons que vous spécifiez trois colonnes, dont chaque ligne contient l'une des deux valeurs possibles. La `Dept` colonne peut avoir la valeur `Admin` ou`Eng`. La `Staff-type` colonne peut avoir la valeur `Part-time` ou`Full-time`. La `Location` colonne peut avoir la valeur `Office1` ou`Office2`. Les compartiments Amazon S3 pour le résultat de votre travail ressemblent à ce qui suit.

```
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
```

Automatiser l'exécution des tâches selon un calendrier

Vous pouvez réexécuter les DataBrew tâches à tout moment et automatiser les exécutions de DataBrew tâches selon un calendrier.

Pour réexécuter une tâche DataBrew

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.
2. Dans le volet de navigation, sélectionnez Jobs. Choisissez le travail que vous souhaitez exécuter, puis sélectionnez Exécuter le travail.

Pour exécuter une DataBrew tâche à un moment précis ou de manière récurrente, créez un calendrier de DataBrew tâches. Vous pouvez ensuite configurer votre tâche pour qu'elle s'exécute conformément au calendrier.

Pour créer un calendrier de DataBrew travail

1. Dans le volet de navigation de la DataBrew console, sélectionnez Jobs. Cliquez sur l'onglet Programmes, puis sélectionnez Ajouter un calendrier.
2. Entrez un nom pour votre calendrier, puis choisissez une valeur pour Fréquence d'exécution :
   - Récurrent : choisissez la fréquence à laquelle vous souhaitez que le travail soit exécuté (par exemple, toutes les 12 heures). Choisissez ensuite le ou les jours où vous souhaitez exécuter la tâche. Vous pouvez éventuellement saisir l'heure à laquelle la tâche est exécutée.
   - À une heure précise : entrez l'heure à laquelle vous souhaitez que la tâche soit exécutée. Choisissez ensuite le ou les jours où vous souhaitez exécuter la tâche.
   - Entrez CRON : définissez le calendrier des tâches en saisissant une expression cron valide. Pour de plus amples informations, veuillez consulter Utilisation d'expressions cron pour les tâches de recette.
3. Lorsque les paramètres vous conviennent, choisissez Enregistrer.

Pour associer une tâche à un planning

1. Dans le volet de navigation, sélectionnez Jobs.
2. Choisissez le travail sur lequel vous souhaitez travailler, puis dans Actions, sélectionnez Modifier. .
3. Dans le volet Planifier les tâches, choisissez Associer la planification. Choisissez le nom du calendrier que vous souhaitez utiliser.
4. Lorsque les paramètres vous conviennent, choisissez Enregistrer.

Utilisation d'expressions cron pour les tâches de recette

Ces expressions se composent de six champs obligatoires qui sont séparés par des espaces. La syntaxe est la suivante.

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

Dans la syntaxe précédente, les valeurs et caractères génériques suivants sont utilisés pour les champs indiqués.

Champs	Valeurs	Caractères génériques
Minutes	0–59	, - * /
Heures	0 – 23	, - * /
Day-of-month	1–31	, - * ? / L W
Mois	1 à 12 ou JAN-DEC	, - * /
Day-of-week	1 à 7 ou SUN-SAT	, - * ? / L
Année	1970-2199	, - * /

Utilisez ces caractères génériques comme suit :

- Le caractère générique , (virgule) inclut des valeurs supplémentaires. Sur le `Month` terrain, cela `JAN,FEB,MAR` inclut les mois de janvier, février et mars.
- Le caractère générique - (en tiret) indique les plages. Sur le `Day` terrain, 1 à 15 inclut les jours 1 à 15 du mois spécifié.
- Le caractère générique * (astérisque) inclut toutes les valeurs du champ. Sur le `Hours` terrain, * inclut toutes les heures.
- Le caractère générique / (barre oblique) spécifie les incréments. Dans le `Minutes` champ, vous pouvez saisir **`1/10`** pour spécifier toutes les 10 minutes, à partir de la première minute de l'heure (par exemple, les 11, 21 et 31 minutes).

- Le caractère générique ? (point d'interrogation) indique l'un ou l'autre. Supposons, par exemple, que vous saisissiez 7 dans le `Day-of-month` champ. Si vous ne vous souciez pas du jour de la semaine le 7, pouvez-vous alors participer ? sur le `Day-of-week` terrain.
- Le caractère générique L dans le `Day-of-week` champ `Day-of-month` ou indique le dernier jour du mois ou de la semaine.
- Le caractère générique W dans le champ spécifie un jour de la semaine. `Day-of-month` Dans le champ `Day-of-month`, `3W` spécifie le jour le plus proche du troisième jour de semaine du mois.

Ces champs et valeurs présentent les limites suivantes :

- Vous ne pouvez pas spécifier les champs `Day-of-month` et `Day-of-week` de la même expression cron. Si vous spécifiez une valeur dans l'un de ces champs, vous devez utiliser un signe ? (point d'interrogation) dans l'autre.
- Les expressions Cron qui génèrent des débits supérieurs à 5 minutes ne sont pas prises en charge.

Lors de la création d'une planification, vous pouvez utiliser les exemples de chaînes cron suivants.

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0 USD	10	*	*	?	*	Fonctionne à 10 h 00 (UTC) tous les jours
15	12	*	*	?	*	Exécuter à 12 h 15 (UTC) chaque jour
0	18	?	*	MON-FRI	*	Exécuter à 18 h 00 (UTC) du

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
						lundi au vendredi
0	8	1	*	?	*	Ouvert à 8 h 00 (UTC) tous les premiers jours du mois
0/15	*	*	*	?	*	Exécuter toutes les 15 minutes
0/10	*	?	*	MON-FRI	*	Exécuter toutes les 10 minutes du lundi au vendredi
0/5	8–17	?	*	MON-FRI	*	Exécuter toutes les 5 minutes du lundi au vendredi entre 8 h 00 et 17 h 55 (UTC)

Par exemple, vous pouvez utiliser l'expression cron suivante pour exécuter une tâche tous les jours à 12 h 15 UTC.

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

Supprimer des tâches et des plannings de tâches

Si vous n'avez plus besoin d'un travail ou d'un calendrier de travail, vous pouvez le supprimer.

Pour supprimer une tâche

1. Dans le volet de navigation, sélectionnez Jobs.
2. Choisissez le travail que vous souhaitez supprimer, puis dans Actions, choisissez Supprimer. .

Pour supprimer un planning de travail

1. Dans le volet de navigation, choisissez Jobs, puis sélectionnez l'onglet Schedules.
2. Choisissez le calendrier que vous souhaitez supprimer, puis dans Actions, choisissez Supprimer. .

Création et utilisation de AWS Glue DataBrew emplois de profil

Les tâches de profilage exécutent une série d'évaluations sur un ensemble de données et transmettent les résultats à Amazon S3. Les informations collectées par le profilage des données vous aident à comprendre votre ensemble de données et à décider du type d'étapes de préparation des données que vous souhaiterez peut-être exécuter dans vos tâches de recette.

La méthode la plus simple pour exécuter une tâche de profilage consiste à utiliser les DataBrew paramètres par défaut. Vous pouvez configurer votre tâche de profil avant de l'exécuter afin qu'elle renvoie uniquement les informations souhaitées.

Utilisez la procédure suivante pour créer une tâche DataBrew de profil.

Pour créer un profil d'emploi

1. Connectez-vous à la DataBrew console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/databrew/.
2. Choisissez JOBS dans le volet de navigation, choisissez l'onglet Profile jobs, puis choisissez Create job.
3. Entrez un nom pour votre travail, puis choisissez Créer un profil d'emploi.
4. Pour la saisie Job, indiquez le nom de l'ensemble de données à profiler.

5. (Facultatif) Configurez les éléments suivants dans le volet de configuration des profils de données :

   - Configurations au niveau du jeu de données : configurez les détails de votre tâche de profil pour toutes les colonnes de votre ensemble de données.

     Vous pouvez éventuellement activer la capacité de détecter et de compter les lignes dupliquées dans le jeu de données. Vous pouvez également choisir Activer la matrice de corrélations et sélectionner des colonnes pour voir dans quelle mesure les valeurs de plusieurs colonnes sont étroitement liées. Pour plus de détails sur les statistiques que vous pouvez configurer au niveau du jeu de données, consultezStatistiques configurables au niveau du jeu de données. Vous pouvez configurer les statistiques sur la DataBrew console ou à l'aide de l' DataBrewAPI ou AWS des SDK.

   - Configurations au niveau des colonnes : à l'aide des paramètres de configuration de profil par défaut, vous pouvez sélectionner les colonnes à inclure dans votre tâche de profil. Utilisez Ajouter un remplacement de configuration pour sélectionner les colonnes pour lesquelles vous souhaitez limiter le nombre de statistiques collectées ou remplacer la configuration par défaut de certaines statistiques. Pour plus de détails sur les statistiques que vous pouvez configurer au niveau des colonnes, consultezStatistiques configurables au niveau des colonnes. Vous pouvez configurer les statistiques sur la DataBrew console ou à l'aide de l' DataBrew API ou AWS des SDK.

     Assurez-vous que les remplacements de configuration que vous spécifiez s'appliquent aux colonnes que vous avez incluses dans votre profil de travail. En cas de conflit entre les différents remplacements que vous avez configurés pour une colonne, le dernier remplacement conflictuel est prioritaire.

6. (Facultatif) Vous pouvez créer des règles de qualité des données et appliquer des ensembles de règles supplémentaires associés à cet ensemble de données ou supprimer des règles déjà appliquées. Pour plus d'informations sur la validation de la qualité des données, voirValidation de la qualité des données dans AWS Glue DataBrew.

7. Dans le volet Paramètres avancés des tâches, vous pouvez choisir d'autres options concernant le mode d'exécution de votre tâche :

   - Nombre maximum d'unités : DataBrew traite les tâches à l'aide de plusieurs nœuds de calcul et s'exécute en parallèle. Le nombre de nœuds par défaut est de 5. Le nombre maximum de nœuds est de 149.

- Délai d'expiration d'une tâche : si une tâche prend plus de minutes que le nombre de minutes que vous avez défini ici pour s'exécuter, elle échoue avec une erreur de temporisation. La valeur par défaut est de 2 880 minutes, soit 48 heures.
- Nombre de tentatives : si une tâche échoue en cours d'exécution, DataBrew vous pouvez essayer de l'exécuter à nouveau. Par défaut, la tâche n'est pas réessayée.
- Activer Amazon CloudWatch Logs pour le travail : permet DataBrew de publier des informations de diagnostic dans CloudWatch Logs. Ces journaux peuvent être utiles à des fins de résolution des problèmes ou pour obtenir plus de détails sur le traitement de la tâche.

8. Pour Associated Schedule, vous pouvez appliquer un calendrier de DataBrew travail afin que votre travail soit exécuté à un moment précis ou de manière récurrente. Pour de plus amples informations, veuillez consulter Automatiser l'exécution des tâches selon un calendrier.
9. Lorsque les paramètres sont tels que vous le souhaitez, choisissez Create job. Ou, si vous souhaitez exécuter le travail immédiatement, choisissez Create and run job.

Création d'une configuration de tâche de profil par programmation dans AWS Glue DataBrew

Dans cette section, vous trouverez des descriptions des étapes et des fonctions des tâches de profilage que vous pouvez utiliser par programmation. Vous pouvez les utiliser soit à partir du AWS Command Line Interface(AWS CLI), soit à l'aide de l'un des AWS SDK.

Dans une tâche de profilage, vous pouvez personnaliser une configuration pour contrôler le mode d' DataBrew évaluation de votre ensemble de données. Vous pouvez appliquer la configuration à un ensemble de données ou à des colonnes spécifiques. Vous pouvez créer la configuration lors de la création d'une tâche de profil, puis la mettre à jour à tout moment.

Une structure de configuration de profil comprend quatre parties :

- ProfileColumns section
- DatasetStatisticsConfiguration section
- ColumnStatisticsConfigurations section
- EntityDetectorConfiguration section pour configurer les informations personnelles

Voici un exemple.

```
{
```

```
    "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 section

Dans la `ProfileColumns` section de votre structure, définissez les colonnes de votre jeu de données que vous souhaitez évaluer dans votre profil de travail. `ProfileColumns`est une liste de sélecteurs de colonnes (`Selectors`). Vous pouvez spécifier un nom de colonne ou une expression régulière dans un sélecteur de colonne. Un exemple suit.

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

Lorsque cette `ProfileColumns` option est spécifiée, seules les colonnes dont le nom correspond à un nom ou à une expression régulière `ProfileColumns` sont incluses dans la tâche de profilage. Si le travail de profil ne prend pas en charge le type de données d'une colonne sélectionnée, DataBrew ignore la colonne sélectionnée pendant l'exécution du travail.

S'il ProfileColumns n'est pas défini, la tâche de profilage évalue toutes les colonnes prises en charge. Les colonnes prises en charge sont des colonnes contenant des données d'un type de données pris en charge : `ByteType` `ShortType` `IntegerTypeLongType`,`FloatType`,`DoubleType`,`String`,, ou`Boolean`.

DatasetStatisticsConfiguration section

Dans la `DatasetStatisticsConfiguration` section de votre structure, vous pouvez créer une configuration pour les évaluations intercolonnes. La configuration inclut `IncludedStatistics` et0`verrides`. Un exemple suit.

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

```
}
```

Vous pouvez sélectionner les évaluations que vous souhaitez avoir en y ajoutant des noms d'évaluation`IncludedStatistics`. Un exemple suit.

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

Lorsque vous le spécifiez`IncludedStatistics`, seules les évaluations de la liste sont incluses dans le profil du poste. S'il `IncludedStatistics` n'est pas défini, le job de profilage exécute toutes les évaluations prises en charge avec les paramètres par défaut. Vous pouvez exclure toutes les évaluations en ajoutant AUCUNE à`IncludedStatistics`. Un exemple suit.

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

Statistiques configurables au niveau du jeu de données

Dans la `DatasetStatisticsConfiguration` section de votre structure, un poste de profil soutient les évaluations indiquées dans le tableau suivant.

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
NOMBRE DE LIGNES DUPLIQUÉES	Nombre de lignes dupliquées dans le jeu de données	tout	Enable	dupliquer RowsCoun	Int
CORRÉLATION	Coefficient de corrélation de Pearson entre deux colonnes	number	Enable	corrélations (dans chaque colonne sélectionnée)	Objet

Dans`IncludedStatistics`, vous pouvez remplacer les paramètres par défaut de chaque évaluation en ajoutant une dérogation. Chaque dérogation inclut le nom d'une évaluation particulière et une carte de paramètres.

Dans`DatasetStatisticsConfiguration`, une tâche de profil prend en charge la `CORRELATION` dérogation. Cette dérogation calcule le coefficient de corrélation de Pearson entre deux colonnes à partir d'une liste de colonnes sélectionnées. Le paramètre par défaut consiste à sélectionner les 10 premières colonnes numériques. Vous pouvez spécifier un certain nombre de colonnes ou une liste de sélecteurs de colonnes pour remplacer le paramètre par défaut.

`CORRELATION`prend les paramètres suivants :

- `columnNumber`— Le nombre de colonnes numériques. La tâche de profilage sélectionne les n premières colonnes de l'ensemble de données. Cette valeur doit être supérieure à 1. Permet `"ALL"` de sélectionner toutes les colonnes numériques.
- `columnSelectors:`— Liste des sélecteurs de colonnes. Chaque sélecteur peut avoir un nom de colonne ou une expression régulière.

Un exemple suit.

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

ColumnStatisticsConfigurations section

Dans la `ColumnStatisticsConfigurations` section de votre structure, vous pouvez créer des configurations pour des colonnes spécifiques. `ColumnStatisticsConfigurations`est une liste de `ColumnStatisticsConfiguration` paramètres. Il y `Selectors` a une liste de sélecteurs de colonnes, et `Statistics` pour la configuration des statistiques. `ColumnStatisticsConfiguration` Un exemple suit.

```
{
    "Selectors": [{"Name": "example"}
    ],
    "Statistics": {
```

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

`Selectors`est une liste de sélecteurs de colonnes. De même`ProfileColumns`, vous pouvez spécifier un nom de colonne ou une expression régulière dans chaque sélecteur de colonne. Lorsque vous le spécifiez`Selectors`, la configuration des colonnes est appliquée aux colonnes qui correspondent à n'importe quel sélecteur de colonne dans`Selectors`. Dans le cas contraire, la configuration est appliquée à toutes les colonnes prises en charge.

Dans`Statistics`, vous pouvez remplacer les paramètres des colonnes sélectionnées. Comme avec`DatasetStatisticsConfiguration`, `Statistics` a `IncludedStatistics` et`Overrides`.

Pour sélectionner les évaluations que vous souhaitez, ajoutez des noms d'évaluation à`IncludedStatistics`.

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

Lorsque vous le spécifiez`IncludedStatistics`, seules les évaluations de la liste sont incluses dans le profil du poste. Dans le cas contraire, la tâche de profilage exécute toutes les évaluations prises en charge avec les paramètres par défaut.

Vous pouvez exclure toutes les évaluations en ajoutant `NONE` à`IncludedStatistics`.

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

Dans certains cas, il peut y avoir plusieurs configurations différentes `IncludedStatistics` que vous pouvez appliquer à la même colonne. `ColumnStatisticsConfigurations` Dans ces cas, la tâche de profilage sélectionne la dernière configuration `ColumnStatisticsConfigurations`

et l'applique `IncludedStatistics` à la colonne sélectionnée. Une nouvelle configuration remplace les anciennes.

Statistiques configurables au niveau des colonnes

En`ColumnStatisticsConfigurations`, un poste profilé soutient les évaluations présentées dans le tableau suivant.

Un type de données pris en charge `number` dans ce tableau signifie que le type de données de l'attribut est l'un des suivants : `ByteTypeShortType`,`IntegerType`,`LongType`,`FloatType`, ou`DoubleType`.

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
–	Nom de la colonne.	tout	–	name	chaîne
–	Type de données de la colonne.	tout	–	type	chaîne
NOMBRE_VALEUR_DISTINCT	Nombre de valeurs distinctes. Une valeur distincte est une valeur qui apparaît au moins une fois.	number/boolean/chaîne	Activé	distinctValuesCount	Int
ENTROPIE	Entropie (théorie de l'information).	number/boolean/chaîne	Activé	entropie	Double
PLAGE INTERQUARTILE	Variez entre 25 % et 75 % des chiffres.	number	Activé	Intervalle interquartile	Double
KURTOSIS	Kurtosis de la colonne.	number	Activé	kurtosis	Double

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
MAX	Valeur maximale dans la colonne.	number/string longueur	Activé	max	Int/Double
VALEUR_MAXIMALES	Liste des valeurs maximales de la colonne et de leur nombre.	number	Activé	Valeurs maximales	List
MEAN	Valeur moyenne des valeurs de la colonne.	number/string longueur	Activé	mean	Double
MEDIAN	Médiane des valeurs de la colonne.	number/string longueur	Activé	median	Double
DÉVIATION ABSOLUE MÉDIANE	La médiane des différences absolues entre chaque point de données et la médiane d'une colonne numérique.	number	Activé	médiane AbsoluteDeviation	Double
MIN	Valeur minimale dans la colonne.	number/string longueur	Activé	min	Int/Double
VALEUR_MINIMALES	Liste des valeurs minimales de la colonne et de leur nombre.	number	Activé	Valeurs minimales	List

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
COMPTE_VALEUR_MANQUANTES	Nombre de valeurs manquantes dans la colonne. Les chaînes nulles et vides sont considérées comme manquantes.	tout	Activé	manquantValuesCount	Int
MODE	La valeur la plus fréquente dans la colonne. Si plusieurs valeurs apparaissent aussi souvent, le mode est l'une de ces valeurs.	number/string longueur	Activé	mode	Int/Double
VALEUR_LES PLUS COMMUNES	Liste des valeurs les plus courantes de la colonne.	number/boolean/chaîne	Activé	le plus CommonValues	List

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
DÉTECTION DES VALEURS ABERRANTES	Détectez les valeurs aberrantes dans la colonne à l'aide de l'algorithme Z_score. Comptez le nombre de valeurs aberrantes et extrayez une liste d'échantillons à partir des valeurs aberrantes détectées.	number/string longueur	Activé	zScoreOutliersCount, zScoreOutliersSample	Int/List
PERCENTILES	Valeurs percentiles de la colonne numérique (5 %, 25 %, 75 %, 95 %).	number	Activé	percentile 5, percentile 25, percentile 75, percentile 95	Double
RANGE	Plage de valeurs dans la colonne.	number	Activé	range	Int/Double
ASYMÉTRIE	Asymétrie des valeurs dans la colonne.	number	Activé	asymétrie	Double

Nom de la statistique	Description	Types de données pris en charge	État par défaut	Attributs du résultat du profil	Type de résultat de profil
ÉCART-TYPE	Écart type d'échantillon non biaisé des valeurs de la colonne.	number/string longueur	Activé	Écart type	Double
SUM	Somme des valeurs de la colonne.	number	Activé	sum	Int/Double
DÉNOMBRE_VALEUR_UNIQUE	Nombre de valeurs uniques. Une valeur unique signifie qu'elle n'apparaît qu'une seule fois.	number/boolean/chaîne	Activé	uniqueValuesCount	Int
DISTRIBUTION_VALEUR	Mesure de la distribution des valeurs dans la colonne par plage.	number/string longueur	Activé	Répartition de la valeur	List
ÉCART	Variance des valeurs de la colonne.	number	Activé	variance	Double
Z_SCORE_DISTRIBUTION	Mesure de la distribution des valeurs du score Z des points de données par plage.	number	Activé	zScoreDistribution	List
NOMBRE DE ZÉROS	Nombre de zéros (0) dans la colonne.	number	Activé	Nombre de zéros	Int

Dans`IncludedStatistics`, vous pouvez remplacer les paramètres par défaut de chaque évaluation en ajoutant une dérogation. Chaque dérogation inclut le nom d'une évaluation particulière et une carte de paramètres.

Paramètres des ColumnStatisticsConfigurations colonnes

Dans`ColumnStatisticsConfigurations`, une tâche de profilage prend en charge les paramètres suivants.

Dans certains cas, il peut y avoir plusieurs configurations différentes `IncludedStatistics` que vous pouvez appliquer à la même colonne. `ColumnStatisticsConfigurations` Dans ces cas, la tâche de profilage sélectionne la dernière configuration `ColumnStatisticsConfigurations` et l'applique `IncludedStatistics` à la colonne sélectionnée. Une nouvelle configuration remplace les anciennes.

VALEUR_MAXIMALES

Répertorie les valeurs maximales de la colonne numérique et leur nombre. La taille de liste par défaut est 5. Vous pouvez modifier la taille de la liste en spécifiant une valeur pour`sampleSize`.

Paramètres

`sampleSize`— La taille de la liste qui inclut le nombre et le nombre maximum de valeurs dans la colonne numérique. Cette valeur doit être supérieure à 0. Permet `"ALL"` de répertorier toutes les valeurs.

Exemple

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

VALEUR_MINIMALES

Répertorie les valeurs minimales de la colonne numérique et leur nombre. La taille de liste par défaut est 5. Vous pouvez modifier la taille de la liste en spécifiant une valeur pour`sampleSize`.

Paramètres

`sampleSize`— La taille de la liste qui inclut le nombre et le nombre maximum de valeurs dans la colonne numérique. Cette valeur doit être supérieure à 0. Permet `"ALL"` de répertorier toutes les valeurs.

Exemple

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

VALEUR_LES PLUS COMMUNES

Répertorie les valeurs les plus courantes de la colonne et leur nombre. La taille de liste par défaut est 50. Vous pouvez modifier la taille de la liste en spécifiant une valeur pour`sampleSize`.

Paramètres

`sampleSize`— La taille de la liste qui inclut le nombre et le nombre maximum de valeurs dans la colonne numérique. Cette valeur doit être supérieure à 0. Permet `"ALL"` de répertorier toutes les valeurs.

Exemple

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

DÉTECTION DES VALEURS ABERRANTES

Détecte les valeurs aberrantes dans la colonne numérique ou la colonne de chaîne (en fonction de la longueur de la chaîne) par l'algorithme Z_score.

Votre job de profil compte le nombre de valeurs aberrantes et génère une liste d'échantillons de valeurs aberrantes et de leurs scores Z. La liste d'échantillons est ordonnée en fonction de la valeur absolue du score Z. La taille de liste par défaut est 50.

L'algorithme Z_Score identifie une valeur comme une valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type. Le seuil de valeur aberrante par défaut est 3.

Vous pouvez fournir un seuil supplémentaire, un seuil modéré, pour obtenir plus d'informations. Votre seuil modéré doit être inférieur à votre seuil. Cette fonctionnalité est désactivée par défaut. Lorsqu'un seuil modéré est spécifié, votre profil d'emploi renvoie un compte de plus,`zScoreMildOutliersCount`. `zScoreOutliersSample`Peut également inclure un échantillon de seuils légèrement aberrants dans ce cas.

Paramètres

- `threshold`— La valeur seuil à utiliser lors de la détection des valeurs aberrantes. Cette valeur doit être supérieure ou égale à 0.
- `mildThreshold`— La valeur seuil modérée à utiliser lors de la détection des valeurs aberrantes. Cette valeur doit être supérieure ou égale à 0 et inférieure à`threshold`.
- `sampleSize`— Taille de la liste qui inclut les valeurs aberrantes dans la colonne. Permet `"ALL"` de répertorier toutes les valeurs.

Exemple

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

DISTRIBUTION_VALEUR

Mesure la distribution des valeurs dans la colonne en fonction des plages de valeurs. Une tâche de profilage regroupe les valeurs d'une colonne numérique ou d'une colonne de chaîne (en fonction de

la longueur de la chaîne) dans des groupes par plages numériques et génère une liste de groupes. Les compartiments sont consécutifs, et la limite supérieure d'un compartiment est la limite inférieure du compartiment suivant.

Paramètres

`binNumber`— Nombre de bacs. Cette valeur doit être supérieure à 0.

Exemple

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

Z_SCORE_DISTRIBUTION

Mesure la distribution des scores Z des valeurs dans une colonne numérique. Une tâche de profilage regroupe les scores Z de valeurs dans des groupes par plages numériques et génère une liste de groupes. Les compartiments sont consécutifs, et la limite supérieure d'un compartiment est la limite inférieure du compartiment suivant.

Paramètres

`binNumber`— Nombre de bacs. Cette valeur doit être supérieure à 0.

Exemple

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

EntityDetectorConfiguration section pour configurer les informations personnelles

Dans la `EntityDetectorConfiguration` section de votre structure, vous pouvez configurer les types d'entités de votre ensemble de données que vous DataBrew souhaitez détecter en tant qu'informations personnelles identifiables (PII) pour un poste de profil.

EntityTypes

Vous configurez les types d'entités que vous DataBrew souhaitez détecter en tant que PII pour votre tâche de profil. Lorsqu'elle `EntityDetectorConfiguration` n'est pas définie, la détection des entités est désactivée. Les types d'entités suivants peuvent être détectés dans votre ensemble de données :

- 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

Le groupe de types d'entités `USA_ALL` est également pris en charge et inclut tous les types d'entités ci-dessus, à l'exception de `PERSON_NAME` et`DATE`.

Le type de `EntityTypes` est un tableau de chaînes.

AllowedStatistics

Configurez les statistiques autorisées à être exécutées sur les colonnes contenant les entités détectées. S'il n'`AllowedStatistics`est pas défini, aucune statistique ne sera calculée sur les colonnes contenant les entités détectées. Consultez Statistiques configurables au niveau des colonnes la liste des valeurs valides pour le `AllowedStatistics` paramètre.

Le type de `AllowedStatistics` est un tableau d'`AllowedStatistics`objets.

Sécurité dans AWS Glue DataBrew

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud :AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud.AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Third-partyles auditeurs testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent àAWS Glue DataBrew, consultez les AWS services de la section Étendue par programme de conformité et .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS Glue DataBrew. Les rubriques suivantes expliquent comment procéder à la configuration DataBrew pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos DataBrew ressources.

Rubriques

Protection des données dans AWS Glue DataBrew

DataBrew propose plusieurs fonctionnalités conçues pour protéger vos données.

Rubriques

Le modèle de responsabilité partagée AWS s’applique à la protection des données dans AWS Glue DataBrew. Comme décrit dans ce modèle,AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la FAQ sur la confidentialité des données et les . Pour plus d’informations sur la protection des données en Europe, consultez le Centre du règlement général sur la protection des données (RGPD).

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS(IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l’authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.

- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec DataBrew ou d'autres Services AWS utilisateurs de la console, de l'API ou AWS des SDK.AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement au repos

DataBrew prend en charge le chiffrement des données au repos pour les DataBrew projets et les tâches. Les projets et les tâches peuvent lire des données cryptées, et les tâches peuvent écrire des données chiffrées en appelant AWS Key Management Service(AWS KMS) pour générer des clés et déchiffrer les données. Vous pouvez également utiliser des clés KMS pour chiffrer les journaux de tâches générés par les DataBrew tâches. Vous pouvez spécifier des clés de chiffrement à l'aide de la DataBrew console ou de l' DataBrew API.

> **Important**
>
> AWS Glue DataBrew ne prend en charge que les clés AWS KMS symétriques. Pour plus d'informations, consultez la section Clés AWS KMS dans le Guide du AWS Key Management Service développeur.

Lorsque vous créez des tâches DataBrew avec le chiffrement activé, vous pouvez utiliser la DataBrew console pour spécifier les clés de chiffrement S3-managed côté serveur (SSE-S3) ou les clés KMS stockées dans AWS KMS(SSE-KMS) pour chiffrer les données au repos.

> **Important**
>
> Lorsque vous utilisez un ensemble de données Amazon Redshift, les objets déchargés dans le répertoire temporaire fourni sont chiffrés avec. SSE-S3

Chiffrer les données écrites par les jobs DataBrew

DataBrew les tâches peuvent écrire sur des cibles Amazon S3 chiffrées et sur Amazon CloudWatch Logs chiffrés.

Rubriques

- Configuration DataBrew pour utiliser le chiffrement
- Création d'un itinéraire vers AWS KMS pour les tâches VPC
- Configuration du chiffrement avec AWS Clés KMS

Configuration DataBrew pour utiliser le chiffrement

Suivez cette procédure pour configurer votre DataBrew environnement afin qu'il utilise le chiffrement.

Pour configurer votre DataBrew environnement afin d'utiliser le chiffrement

1. Créez ou mettez à jour vos clés AWS KMS pour accorder des AWS KMS autorisations aux rôles Gestion des identités et des accès AWS(IAM) transmis aux DataBrew tâches. Ces rôles IAM sont utilisés pour chiffrer les CloudWatch journaux et les cibles Amazon S3. Pour plus d'informations, consultez la section Chiffrer les données de journal dans CloudWatch les journaux à l'aide AWS KMS du guide de l'utilisateur Amazon CloudWatch Logs.

   Dans l'exemple suivant, *"role1""role2"*, et *"role3"* sont des rôles IAM transmis à des DataBrew tâches. Cette déclaration de politique décrit une politique de clé KMS qui autorise les rôles IAM répertoriés à chiffrer et à déchiffrer avec cette clé 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": "*"
    }
```

L'`Service`instruction, représentée sous la forme`"Service": "logs.region.amazonaws.com"`, est obligatoire si vous utilisez la clé pour chiffrer CloudWatch les journaux.

2. Assurez-vous que la AWS KMS clé est réglée sur `ENABLED` avant de l'utiliser.

Pour plus d'informations sur la spécification des autorisations à l'aide de politiques AWS KMS clés, consultez la section Utilisation de politiques clés dans AWS KMS.

Création d'un itinéraire vers AWS KMS pour les tâches VPC

Vous pouvez vous connecter directement àAWS KMS via un point de terminaison privé dans votre VPC au lieu de vous connecter via Internet. Lorsque vous utilisez un point de terminaison VPC, la communication entre votre VPC et celui-ci AWS KMS s'effectue entièrement au sein du réseau.AWS

Vous pouvez créer un point de terminaison AWS KMS VPC au sein d'un VPC. Sans cette étape, vos DataBrew tâches risquent d'échouer avec un`kms timeout`. Pour obtenir des instructions détaillées, consultez la section Connexion AWS KMS via un point de terminaison VPC dans le guide du AWS Key Management Service développeur.

Lorsque vous suivez ces instructions, sur la console VPC, veillez à effectuer les opérations suivantes :

- Choisissez Activer le nom DNS privé.

- Pour Groupe de sécurité, choisissez le groupe de sécurité (y compris une règle d'autoréférencement) que vous utilisez pour votre DataBrew tâche qui accède à Java Database Connectivity (JDBC).

Lorsque vous exécutez une DataBrew tâche qui accède aux magasins de données JDBC, vous DataBrew devez disposer d'un itinéraire vers le point de terminaison.AWS KMS Vous pouvez fournir l'itinéraire à l'aide d'une passerelle de traduction d'adresses réseau (NAT) ou d'un point de AWS KMS terminaison VPC. Pour créer une passerelle NAT, veuillez consulter Passerelles NAT dans le Guide de l'utilisateur Amazon VPC.

Configuration du chiffrement avec AWS Clés KMS

Lorsque vous activez le chiffrement sur une tâche, il s'applique à la fois à Amazon S3 et CloudWatch. Le rôle IAM transmis doit disposer des AWS KMS autorisations suivantes.

Pour en savoir plus, consulter les rubriques suivantes dans le Guide de l'utilisateur d'Amazon Simple Storage Service :

- Pour plus d'informations`SSE-S3`, consultez la section Protection des données à l'aide du Server-Side chiffrement avec Amazon S3-Managed Encryption Keys (SSE-S3).
- Pour plus d'informations`SSE-KMS`, voir Protection des données à l'aide du Server-Side chiffrement à l'aide de clés AWS gérées par KMS (). SSE-KMS

Chiffrement en transit

AWS fournit un cryptage SSL (Secure Sockets Layer) pour les données en vol.

DataBrew le support pour les sources de données JDBC est assuré.AWS Glue Lorsque vous vous connectez à des sources de données JDBC, DataBrew utilise les paramètres de votre AWS Glue connexion, notamment l'option Exiger une connexion SSL. Pour plus d'informations, consultez la section Propriétés de AWS Glue connexion -AWS Glue dans le guide du AWS Glue développeur.

AWS KMS fournit à la fois un chiffrement « apportez votre propre clé » et un chiffrement côté serveur pour le traitement de l' DataBrew extraction, de la transformation, du chargement (ETL) et pour le.AWS Glue Data Catalog

Gestion des clés

Vous pouvez utiliser IAM DataBrew pour définir des utilisateurs,AWS des ressources, des groupes, des rôles et des politiques précises concernant l'accès, le refus, etc.

Vous pouvez définir l'accès aux métadonnées à l'aide de politiques basées à la fois sur les ressources et sur l'identité, en fonction des besoins de votre organisation. Resource-based les politiques répertorient les principaux auxquels l'accès est autorisé ou refusé à vos ressources, ce qui vous permet de configurer des politiques telles que l'accès entre comptes. Les stratégies d'identité sont spécifiquement associées à des utilisateurs, des groupes et des rôles au sein d'IAM.

DataBrew prend en charge la création de votre propre cryptage AWS KMS key« apportez votre propre clé ». DataBrew fournit également un chiffrement côté serveur à l'aide des clés KMS de AWS KMS for DataBrew jobs.

Identification et traitement des informations personnelles identifiables (PII)

Lorsque vous créez des fonctions analytiques ou des modèles d'apprentissage automatique, vous avez besoin de mesures de protection pour empêcher l'exposition de données d'identification personnelle (PII). Les informations personnelles sont des données personnelles qui peuvent être utilisées pour identifier une personne, telles qu'une adresse, un numéro de compte bancaire ou un numéro de téléphone. Par exemple, lorsque les analystes de données et les scientifiques des données utilisent des ensembles de données pour découvrir des informations démographiques générales, ils ne devraient pas avoir accès aux informations personnelles de personnes spécifiques.

DataBrew fournit des mécanismes de masquage des données pour masquer les données PII pendant le processus de préparation des données. En fonction des besoins de votre organisation, différents mécanismes de rédaction des données personnelles sont disponibles. Vous pouvez masquer les données d'identification personnelle afin que les utilisateurs ne puissent pas les rétablir, ou vous pouvez rendre l'obfuscation réversible.

L'identification et le masquage des données PII DataBrew impliquent la création d'un ensemble de transformations que les clients peuvent utiliser pour supprimer les données PII. Une partie de ce processus consiste à fournir des statistiques et des fonctionnalités de détection des données personnelles dans le tableau de bord de présentation du profil de données de la DataBrew console.

Vous pouvez utiliser les techniques de masquage de données suivantes :

- Substitution — Remplacez les données PII par d'autres valeurs d'apparence authentique.

- Répartition — Répartissez la valeur d'une même colonne dans différentes lignes.
- Chiffrement déterministe : appliquez des algorithmes de chiffrement déterministes aux valeurs des colonnes. Le chiffrement déterministe produit toujours le même texte chiffré pour une valeur.
- Chiffrement probabiliste : appliquez des algorithmes de chiffrement probabiliste aux valeurs des colonnes. Le chiffrement probabiliste produit un texte chiffré différent chaque fois qu'il est appliqué.
- Déchiffrement : déchiffrez les colonnes en fonction des clés de chiffrement.
- Annulation ou suppression : remplacez un champ spécifique par une valeur nulle ou supprimez la colonne.
- Masquage : utilisez le brouillage de caractères ou masquez certaines parties des colonnes.
- Hachage — Appliquez des fonctions de hachage aux valeurs des colonnes.

Pour plus d'informations sur l'utilisation des transformations, voir Étapes de préparation des informations personnelles identifiables (PII). Pour plus d'informations sur l'utilisation des tâches de profil pour détecter les informations personnelles, y compris une liste des types d'entités pouvant être détectés, consultez la EntityDetectorConfiguration section consacrée à la configuration des tâches de profil dans Création d'une configuration de tâche de profil par programmation.

DataBrew dépendance à l'égard d'autres AWS services

Pour utiliser la DataBrew console, vous devez disposer d'un ensemble minimal d'autorisations pour utiliser les DataBrew ressources de votre AWS compte. Outre ces DataBrew autorisations, la console a besoin des autorisations des services suivants :

- CloudWatch Autorise les journaux à afficher les journaux.
- Autorisations IAM pour répertorier et transmettre des rôles.
- Autorisations Amazon EC2 permettant de répertorier les VPC, les sous-réseaux, les groupes de sécurité, les instances et d'autres objets. DataBrew utilise ces autorisations pour configurer des éléments Amazon EC2 tels que des VPC lors de l'exécution de tâches. DataBrew
- Autorisations Amazon S3 pour répertorier les buckets et les objets.
- AWS Glue autorisations pour lire les objets AWS Glue du schéma, tels que les bases de données, les partitions, les tables et les connexions.
- AWS Lake Formation autorisations pour travailler avec les lacs de données de Lake Formation.

Gestion des identités et des accès pour AWS Glue DataBrew

Gestion des identités et des accès AWS(IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser DataBrew les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Authentification par des identités

L'authentification est la façon dont vous vous connectez àAWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM.Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center(IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d’informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l’utilisateur Connexion à AWS.

Pour l'accès par programmation,AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d’informations, consultez Signature AWS Version 4 pour les demandes d’API dans le Guide de l’utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d’utiliser l’utilisateur racine pour vos tâches quotidiennes. Pour les tâches

qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder àAWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle utilisateur à un rôle IAM (console) ou en appelant une opération AWS CLI ou AWS API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant àAWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource.AWSévalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les

politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Identity-based politiques

Identity-based les politiques sont des documents de politique d'autorisation JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Identity-based les politiques peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Resource-based politiques

Resource-based les politiques sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Resource-based les politiques sont des politiques intégrées qui se trouvent dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

DataBrew ne prend pas en charge les politiques basées sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL.AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

DataBrew ne prend pas en charge les ACL.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) : spécifient les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez Politiques de contrôle de service dans le Guide de l'utilisateur AWS Organizations.
- Politiques de contrôle des ressources (RCP) : définissent les autorisations maximales disponibles pour les ressources de votre organisation. Pour plus d'informations, consultez Politiques de contrôle des ressources (RCP) dans le Guide de l'utilisateur AWS Organizations.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

AWS Glue DataBrew and AWS Lake Formation

AWS Glue DataBrew prend en charge AWS Lake Formation les autorisations pour AWS Glue Data Catalog les tables. Lorsqu'un jeu de données utilise une AWS Glue Data Catalog table enregistrée auprès de Lake Formation, le rôle IAM attribué aux projets ou aux tâches doit disposer des autorisations DESCRIBE et SELECT Lake Formation sur la table.

AWS Glue DataBrew prend en charge l'écriture dans AWS Glue Data Catalog des tables basées sur AWS Lake Formation. Lorsqu'une DataBrew tâche utilise un catalogue de données enregistré auprès de Lake Formation, le rôle IAM attribué aux tâches doit disposer des autorisations INSERT, ALTER et DELETE de Lake Formation pour les tables concernées. Le rôle IAM doit disposer

d'`glue:UpdateTable`autorisations, ainsi que d'autorisations relatives à l'emplacement des données associé à la table du catalogue de données.

Comment ?AWS Glue DataBrew fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à DataBrew, vous devez connaître les fonctionnalités IAM disponibles. DataBrew Pour obtenir une vue d'ensemble de la façon dont DataBrew les autres AWS services fonctionnent avec IAM, consultez la section AWS Services qui fonctionnent avec IAM dans le Guide de l'utilisateur d'IAM.

Rubriques

- DataBrew politiques basées sur l'identité
- Resource-based politiques dans DataBrew
- DataBrew Rôles IAM

DataBrew politiques basées sur l'identité

Avec les politiques basées sur l’identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions selon lesquelles les actions sont autorisées ou refusées. DataBrew prend en charge des actions, des ressources et des clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Actions

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. En d'autres termes, une politique AWS JSON peut spécifier quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément Action d'une politique JSON décrit les actions auxquelles vous pouvez autoriser ou refuser l'accès dans une politique. Les actions de politique possèdent généralement le même nom que l’opération d’API AWS associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n’ont pas d’opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d’actions dans une politique afin d’accorder l’autorisation d’exécuter les opérations associées.

Les actions de politique en DataBrew cours utilisent le préfixe suivant avant l'action :`databrew:`. Par exemple, pour accorder à une personne l'autorisation d'exécuter une instance Amazon EC2 avec l'opération d'API `RunInstances` Amazon EC2, vous incluez l'action `ec2:RunInstances` dans sa politique. Les déclarations de politique doivent inclure un `NotAction` élément `Action` ou. DataBrew définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec lui.

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par des virgules, comme suit :

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

Vous pouvez aussi préciser plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante.

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

Pour consulter la liste des DataBrew actions, reportez-vous à la section Actions définies par AWS Glue DataBrew dans le guide de l'utilisateur IAM.

Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Les DataBrew API suivantes ne prennent pas en charge les autorisations au niveau des ressources :

- ListDatasets
- ListJobs
- ListProjects
- ListRecipes

- ListRulesets
- ListSchedules

La ressource du DataBrew jeu de données possède le nom de ressource Amazon (ARN) suivant.

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

Pour plus d'informations sur le format des ARN, consultez Amazon Resource Names (ARN) et AWS Service Namespaces.

Par exemple, pour spécifier l'`i-1234567890abcdef0`instance dans votre instruction, utilisez l'ARN suivant.

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

Pour spécifier toutes les instances qui appartiennent à un compte spécifique, utilisez le caractère générique (*).

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

Vous ne pouvez pas effectuer certaines DataBrew actions, telles que celles relatives à la création de ressources, sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

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

Pour consulter la liste des types de DataBrew ressources et de leurs ARN, consultez la section Ressources définies par AWS Glue DataBrew dans le guide de l'utilisateur IAM. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS Glue DataBrew.

Clés de condition

DataBrew ne fournit aucune clé de condition spécifique au service, mais il prend en charge l'utilisation de certaines clés de condition globales. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Exemples

Pour consulter des exemples de politiques DataBrew basées sur l'identité, consultez. Identity-based exemples de politiques pour AWS Glue DataBrew

Resource-based politiques dans DataBrew

DataBrew ne prend pas en charge les politiques basées sur les ressources.

DataBrew Rôles IAM

Un rôle IAM est une entité de votre AWS compte qui dispose d'autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec DataBrew

Vous pouvez utiliser des informations d’identification temporaires pour vous connecter à l’aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d'AWS STS API telles que AssumeRoleou GetFederationToken.

DataBrew prend en charge l'utilisation d'informations d'identification temporaires.

Service-linked rôles

Service-linked les rôles permettent aux AWS services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Service-linked les rôles apparaissent dans votre compte IAM et appartiennent au service. Un administrateur peut consulter, mais ne peut pas modifier les autorisations concernant les rôles liés à un service.

Choisir un rôle IAM dans DataBrew

Lorsque vous créez une ressource de jeu de données dans DataBrew, vous choisissez un rôle IAM pour autoriser DataBrew l'accès en votre nom. Si vous avez déjà créé un rôle de service ou un rôle lié à un service, il vous DataBrew fournit une liste de rôles parmi lesquels choisir. Assurez-vous de choisir un rôle qui autorise l'accès en lecture à un compartiment ou à une AWS Glue Data Catalog ressource Amazon S3, selon le cas.

Identity-based exemples de politiques pour AWS Glue DataBrew

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources DataBrew . Ils ne peuvent pas non plus effectuer de tâches à l'aide des AWS API AWS Management ConsoleAWS CLI, ou. Un administrateur doit créer des politiques IAM autorisant les utilisateurs et les

rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la DataBrew console
- Autoriser des utilisateurs à afficher leurs propres autorisations
- Gestion des DataBrew ressources en fonction des balises

Bonnes pratiques en matière de politiques

Identity-based les politiques déterminent si quelqu'un peut créer, accéder ou supprimer DataBrew des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes

doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue.Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la DataBrew console

Pour accéder à la AWS Glue DataBrew console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les informations relatives DataBrew aux ressources de votre AWS compte. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionne pas comme prévu pour les utilisateurs ou les rôles soumis à cette politique.

Pour garantir que les utilisateurs et les rôles peuvent utiliser la DataBrew console, associez également la politique AWS gérée suivante aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

```
AWSDataBrewConsoleAccess
```

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' DataBrew API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous tentez d'effectuer.

Autoriser des utilisateurs à afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
        }
    ]
}
```

Gestion des DataBrew ressources en fonction des balises

Vous pouvez utiliser les conditions de votre politique basée sur l'identité pour gérer les DataBrew ressources en fonction de balises, par exemple pour supprimer, mettre à jour ou décrire les ressources. L'exemple suivant montre une politique qui refuse la suppression d'un projet. Toutefois, la suppression n'est refusée que si le tag Owner du projet a la valeur admin. Cette politique accorde également les autorisations nécessaires pour refuser cette action sur la console.

JSON

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

Vous pouvez attacher cette stratégie aux utilisateurs de votre compte. Si un utilisateur nommé richard-roe tente de supprimer un DataBrew projet, la ressource ne doit pas être étiquetée owner=admin ou owner=admin. Dans le cas contraire, l'utilisateur n'est pas autorisé à supprimer le projet. La clé de condition Owner correspond à la fois à Owner et à owner car les noms des clés de condition ne distinguent pas les majuscules et minuscules. Pour plus d'informations, consultez Éléments de politique JSON IAM : Condition dans le Guide de l'utilisateur IAM.

> **Note**
>
> ListDatasets,, ListJobs, ListProjects ListRecipes ListRulesets, et ne prennent ListSchedules pas en charge le contrôle d'accès basé sur des balises.

AWS politiques gérées pour AWS Glue DataBrew

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, voir les politiques AWS gérées dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent parfois des autorisations supplémentaires à une politique AWS gérée pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont plus susceptibles de mettre à jour une politique AWS gérée lorsqu'une nouvelle fonctionnalité est lancée ou lorsque de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre,AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique ReadOnlyAccessAWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir une liste et une description des politiques relatives aux fonctions de travail, voir les politiques AWS gérées pour les fonctions de travail dans le guide de l'utilisateur d'IAM.

DataBrew mises à jour de AWS stratégies gérées

Consultez les détails des mises à jour des politiques AWS gérées DataBrew depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page

Historique du DataBrew document. La politique gérée se trouve sur la console AWS IAM à AwsGlueDataBrewFullAccessPolicyl'adresse.

Modifier	Description	Date
AWSGlueDataBrewServiceRole— L'autorisation de lecture pour AWS Glue a été ajoutée.	Cette mise à jour ajoute`glue:GetCustomEntityType` . Cette autorisation est requise pour exécuter des tâches AWS Glue DataBrew de profil lorsque PII-identification cette option est activée.	20 mars 2024
AWSGlueDataBrewServiceRole- L'autorisation de lecture pour AWS Glue a été ajoutée.	Cette mise à jour ajoute`glue:BatchGetCustomEntityTypes` . Cette autorisation est requise pour exécuter des tâches AWS Glue DataBrew de profil lorsque PII-identification cette option est activée.	9 mai 2022
AwsGlueDataBrewFullAccessPolicy- Les autorisations de lecture pour Amazon Redshift-Data DescribeStatements et Amazon S3 GetLifecycleConfiguration ont été ajoutées.	Cette mise `redshift-data:DescribeStatement` à jour permet de valider votre code SQL lors de la création d'un ensemble de Redshift-based données Amazon. Il permet également d'`s3:GetLifecycleConfiguration` évaluer si le cycle de vie du préfixe de compartiment Amazon S3 que vous fournissez en tant que répertoire temporaire	4 février 2022

Modifier	Description	Date
	est configuré. En outre, cette modification remplace les autorisations « databrew : * » par une liste explicite d'autorisations incluant toutes les DataBrew API.	
AwsGlueDataBrewFullAccessPolicy- Read/write des autorisations pour AWS Secrets Manager ont été ajoutées.	Cette mise à jour ajoute, `secretsmanager:CreateSecret` et `secretsmanager:GetSecretValue` pour un secret nommé`databrew!default` , un secret par défaut à utiliser avec les DataBrew transformations. En outre, il ajoute des autorisations à CreateSecret for secrets préfixées par `AwsGlueDataBrew-` pour créer des secrets depuis la DataBrew console. GenerateRandom, décrit dans la référence de l'AWS Key Management Service API, est utilisé pour générer une chaîne d'octets aléatoire sécurisée sur le plan cryptographique.	18 novembre 2021

Modifier	Description	Date
AWSGlueDataBrewServiceRole- Read/write des autorisations pour AWS Secrets Manager ont été ajoutées.	Cette mise à jour ajoute `secretsmanager:GetSecretValue` un secret nommé`databrew!default` , un secret par défaut à utiliser avec les DataBrew transformations.	18 novembre 2021
AwsGlueDataBrewFullAccessPolicy- Read/write des autorisations pour AWS Secrets Manager ont été ajoutées.	Cette mise à jour ajoute, `secretsmanager:CreateSecret` et `secretsmanager:GetSecretValue` pour un secret nommé`databrew!default` , un secret par défaut à utiliser avec les DataBrew transformations. En outre, il ajoute des autorisations à CreateSecret for secrets préfixées par `AwsGlueDataBrew-` pour créer des secrets depuis la DataBrew console. `kms:GenerateRandom` (`https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html`) est utilisé pour générer une chaîne d'octets aléatoire sécurisée sur le plan cryptographique.	18 novembre 2021

Modifier	Description	Date
AWSGlueDataBrewServiceRole- Read/write des autorisations pour AWS Secrets Manager ont été ajoutées.	Cette mise à jour ajoute `secretsmanager:GetSecretValue` un secret nommé`databrew!default` , un secret par défaut à utiliser avec les DataBrew transformations.	18 novembre 2021
AwsGlueDataBrewFullAccessPolicy- Des autorisations de lecture pour les bases de données de AWS Glue catalogue et des autorisations de création pour la table de AWS Glue catalogue ont été ajoutées.	Cette mise à jour ajoute des autorisations pour répertorier les bases de données du AWS Glue catalogue et créer de nouvelles tables de catalogue dans une base de données existante dans le cadre de la configuration des sorties pour les DataBrew tâches.	30 Juin 2021
AwsGlueDataBrewFullAccessPolicy- Read/write des autorisations pour la fonctionnalité AppFlow de jeu de données Amazon ont été ajoutées.	Cette mise à jour ajoute des autorisations pour lire les AppFlow flux Amazon existants et les exécutions de flux et pour créer des exécutions de flux.	28 avril 2021

Modifier	Description	Date
AwsGlueDataBrewFullAccessPolicy- Des autorisations de lecture pour les ensembles de données de base de données ont été ajoutées.	Cette mise à jour ajoute des autorisations pour lire AWS Glue les connexions existantes et créer de nouvelles AWS Glue connexions à utiliser avec DataBrew. De plus, pour faciliter l'expérience de création de nouvelles connexions sur console, il permet de répertorier les ressources Amazon VPC et les clusters Amazon Redshift. Il donne également l'autorisation de répertorier les AWS Secrets Manager secrets, mais pas de les lire.	30 mars 2021
DataBrew a commencé à suivre les modifications	DataBrew a commencé à suivre les modifications apportées AWSà ses politiques gérées.	30 mars 2021

Résolution des problèmes d'identité et d'accès dans AWS Glue DataBrew

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec DataBrew IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans DataBrew
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte pour accéder à mes DataBrew ressources

Je ne suis pas autorisé à effectuer une action dans DataBrew

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, contactez votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni vos informations de connexion.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` tente d'utiliser la console pour afficher des informations détaillées concernant un projet mais ne dispose pas des autorisations `databrew:DescribeProject`.

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

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-project`* à l'aide de l'action `databrew:`*`GetProject`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à DataBrew.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans DataBrew. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte pour accéder à mes DataBrew ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises DataBrew en charge, consultezComment ?AWS Glue DataBrew fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Connexion et surveillance DataBrew

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité DataBrew et des performances de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant.AWS fournit plusieurs outils pour surveiller vos DataBrew ressources et répondre aux incidents potentiels :

CloudWatch Alarmes Amazon

À l'aide des CloudWatch alarmes Amazon, vous observez une seule métrique sur une période que vous spécifiez. Si la métrique dépasse un seuil donné, une notification est envoyée à une rubrique ou AWS Auto Scalingà une politique Amazon SNS. CloudWatch les alarmes n'appellent pas d'actions car elles sont dans un état particulier. L'état doit avoir changé et avoir été conservé pendant un nombre de périodes spécifié.

AWS CloudTrail Journaux

CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans DataBrew. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite DataBrew, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Validation de conformité pour AWS Glue DataBrew

Third-party les auditeurs évaluent la sécurité et la conformité dans AWS Glue DataBrew le cadre de multiples programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour savoir si un programme Services AWS de conformitéService AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Résilience dans AWS Glue DataBrew

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité.AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les

zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

En AWS Glue DataBrew effet, nous vous suggérons de configurer vos tâches pour qu'elles utilisent une ou plusieurs tentatives. Le nombre de tentatives pour une tâche est configuré dans la DataBrew console sous Paramètres avancés de la tâche.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans AWS Glue DataBrew

Dans le cadre d'un service géré,AWS Glue DataBrew il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder DataBrew via le réseau. Les clients doivent supporter le protocole TLS (Sécurité de la couche transport) 1.0 ou une version ultérieure. Nous vous recommandons le certificat TLS 1.2 ou une version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement à parfaite confidentialité (PFS), telles que Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Rubriques

- Utilisation AWS Glue DataBrew avec votre VPC
- Utilisation AWS Glue DataBrew avec points de terminaison VPC

Utilisation AWS Glue DataBrew avec votre VPC

Si vous utilisez Amazon VPC pour héberger vos AWS ressources, vous pouvez configurer AWS Glue DataBrew pour acheminer le trafic via votre cloud privé virtuel (VPC) en fonction du service

Amazon VPC. DataBrew pour ce faire, configurez d'abord une interface elastic network dans le sous-réseau que vous spécifiez. DataBrew attache ensuite le groupe de sécurité que vous spécifiez à cette interface réseau pour contrôler l'accès. Le groupe de sécurité spécifié doit disposer de règles d'autoréférencement entrant et sortant pour l'ensemble du trafic. En outre, les noms d'hôte et la résolution DNS doivent être activés sur votre VPC. Pour plus d'informations, consultez la section Configuration d'un VPC pour se connecter aux magasins de données JDBC dans le guide du développeur.AWS Glue

Pour les AWS Glue Data Catalog ensembles de données, les informations VPC sont configurées lorsque vous créez AWS Glue une connexion dans le catalogue de données. Pour créer des tables de catalogue de données pour cette connexion, exécutez un robot d'exploration depuis la AWS Glue console. Pour plus d'informations, consultez la section Remplissage AWS Glue Data Catalog du manuel du AWS Glue développeur.

Pour les ensembles de données de base de données, spécifiez vos informations VPC lorsque vous créez la connexion depuis DataBrew la console.

Pour l'utiliser AWS Glue DataBrew avec un sous-réseau VPC sans NAT, vous devez disposer d'un point de terminaison VPC de passerelle vers Amazon S3 et d'un point de terminaison VPC pour l'interface.AWS Glue Pour plus d'informations, consultez Créer un point de terminaison de passerelle et des points de terminaison VPC d'interface (AWS PrivateLink) dans la documentation Amazon VPC. L'interface Elastic mise en service par DataBrew ne possède pas d'adresse IPv4 publique et ne prend donc pas en charge l'utilisation d'une passerelle Internet VPC.

Les points de terminaison de l'interface Amazon S3 ne sont pas pris en charge pour le moment. Si vous avez l'habitude AWS Secrets Manager de stocker votre secret, vous avez besoin d'un itinéraire vers Secrets Manager. Si vous utilisez le chiffrement, vous avez besoin d'un itinéraire vers AWS Key Management Service(AWS KMS).

Utilisation AWS Glue DataBrew avec points de terminaison VPC

Si vous utilisez Amazon VPC pour héberger vos AWS ressources, vous pouvez établir une connexion privée entre votre VPC et en DataBrew provisionnant un point de terminaison VPC. À l'aide de ce point de terminaison VPC, vous pouvez effectuer des appels DataBrew d'API.

Il n'est pas nécessaire d'utiliser un point de terminaison DataBrew VPC DataBrew avec votre VPC. Pour de plus amples informations, veuillez consulter Utilisation AWS Glue DataBrew avec votre VPC.

Vous pouvez l'utiliser AWS Glue avec des points de terminaison VPC dans toutes les AWS régions qui prennent en charge à la fois des points de terminaison VPC et des points de terminaison AWS Glue VPC.

Pour plus d'informations, veuillez consulter les rubriques suivantes dans le Amazon VPC Guide de l'utilisateur :

- Qu’est-ce qu’Amazon VPC ?
- Création d'un point de terminaison d'interface

Analyse de configuration et de vulnérabilité dans AWS Glue DataBrew

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous, notre client. Pour plus d'informations, consultez le modèle de responsabilitéAWS partagée.

Contrôle AWS Glue DataBrew

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilitéAWS Glue DataBrew et des performances de vos autres AWS solutions.AWS fournit les outils de surveillance suivants pour surveiller DataBrew, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos instances Amazon EC2 et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Events vous permet de configurer des notifications automatiques pour des événements spécifiques dans DataBrew. Les événements de DataBrew sont transmis à CloudWatch Events en temps quasi réel. Vous pouvez configurer les CloudWatch événements pour surveiller les événements et appeler des cibles en réponse à des événements indiquant des modifications de vos partages de ressources. Les modifications apportées à un partage de ressources déclenchent des événements à la fois pour le propriétaire du partage de ressources et pour les principaux autorisés à accéder au partage de ressources. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Events.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d'instances Amazon EC2 et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- AWS CloudTrailcapture les appels d'API et les événements connexes effectués par ou au nom de votre AWS compte. Puis, il livre les fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.

Rubriques

- Surveillance DataBrew avec Amazon CloudWatch
- Automatisation DataBrew grâce aux événements CloudWatch
- Surveillance à DataBrew l'aide de CloudWatch journaux
- Journalisation des appels d' DataBrew API avec AWS CloudTrail
- Utilisation AWS Notifications utilisateur avec AWS Glue Brew de données

Surveillance DataBrew avec Amazon CloudWatch

Vous pouvez surveiller DataBrew l'utilisation CloudWatch, qui collecte les données brutes et les transforme en indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

AWS Glue DataBrew affiche les métriques suivantes dans l'espace de `AWS/DataBrew` noms.

Métrique	Description
`SessionCount`	Le nombre total de DataBrew sessions sur le compte du client Dimensions valides : LogGroupName Statistique valide : somme Unités : nombre

Automatisation DataBrew grâce aux événements CloudWatch

Amazon CloudWatch Events vous permet d'automatiser vos AWS services et de répondre automatiquement aux événements du système tels que les problèmes de disponibilité des applications ou les modifications des ressources. Les événements issus AWS des services sont transmis à CloudWatch Events en temps quasi réel. Vous pouvez écrire des règles simples pour indiquer quels événements vous intéressent et les actions automatisées à effectuer quand un

événement correspond à une règle. Les actions pouvant être déclenchées automatiquement sont les suivantes :

- Invocation de la commande d'exécution Amazon EC2
- Relais de l’événement à Amazon Kinesis Data Streams
- Activation d'une machine àAWS Step Functionsétats
- Notification d’une rubrique Amazon SNS ou d’une file d’attente Amazon SQS

DataBrew rapporte un événement à CloudWatch Events chaque fois que l'état d'une ressource de votre AWS compte change. Les événements sont générés dans la mesure du possible.

Vous trouverez ci-dessous des exemples de plusieurs événements illustrant les différents états d'une DataBrew tâche : `SUCCEEDED`,`FAILED`,`TIMEOUT`, et`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"
  }
```

```
}
```

Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Events.

Surveillance à DataBrew l'aide de CloudWatch journaux

Vous pouvez surveiller les DataBrew tâches à l'aide CloudWatch des journaux, qui collectent des informations détaillées à partir du sous-système des DataBrew tâches et les mettent à disposition pour examen. Ces journaux peuvent être utiles si vous souhaitez avoir un aperçu des ressources utilisées par votre profil et vos jobs de recettes, ou à des fins de dépannage. Pour plus d'informations, consultez le guide de l'utilisateur Amazon CloudWatch Logs.

Journalisation des appels d' DataBrew API avec AWS CloudTrail

DataBrew est intégré àAWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans DataBrew. CloudTrail capture tous les appels d'API DataBrew sous forme d'événements. Les appels capturés incluent des appels provenant de la DataBrew console et des appels de code vers les opérations de l' DataBrew API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour DataBrew. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à DataBrew. Vous pouvez aussi déterminer l'adresse IP à partir de laquelle la demande a été faite, qui a effectué la demande, quand elle a eu lieu et autres informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

DataBrew Informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans DataBrew, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour DataBrew, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur AWS CloudTrail :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les DataBrew actions sont enregistrées CloudTrail et documentées dans la référence de l'API. Par exemple, les appels au `CreateDataset` `UpdateRecipe` et les `StartJobRun` actions génèrent des entrées dans les fichiers CloudTrail journaux.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Comprendre les entrées du fichier DataBrew journal

Encore une fois, CloudTrail un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers

journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateProfileJob`opération.


```
{
    "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"
}
```

Utilisation AWS Notifications utilisateur avec AWS Glue Brew de données

Vous pouvez utiliser les notifications AWS utilisateur pour configurer des canaux de diffusion afin d'être informé des événements de AWS Glue Datbrew. Vous recevez une notification lorsqu'un événement correspond à une règle que vous avez spécifiée. Vous pouvez recevoir des notifications relatives à des événements via plusieurs canaux, notamment par email, via des notifications de chat Amazon Q Developer dans les applications de chat ou via des notifications push de l'AWS Console Mobile Application. Vous pouvez également consulter les notifications dans le Centre de notifications de la console.AWS Les notifications utilisateur prennent en charge l'agrégation, ce qui peut réduire le nombre de notifications que vous recevez lors d'événements spécifiques.

Étape de recette et référence des fonctions

Dans cette référence, vous trouverez des descriptions des étapes de la recette et des fonctions que vous pouvez utiliser par programmation, soit à partir du SDK, soit à l'AWS CLI aide de l'un des SDK.AWS Dans DataBrew, une étape de recette est une action qui transforme vos données brutes en un formulaire prêt à être consommé par votre pipeline de données. Une DataBrew fonction est un type spécial d'étape de recette qui effectue un calcul basé sur des paramètres.

Les catégories de transformations dans l'interface utilisateur sont les suivantes :

- Étapes de base de la recette des colonnes
  - Filtre
  - Colonne
- Étapes de la recette de nettoyage des données
  - Format
  - Propre
  - Extrait
- Étapes de la recette de qualité des données
  - Manquant
  - Non valide
  - Duplicates (doublons)
  - Valeurs aberrantes
- Étapes de la recette contenant des informations personnellement identifiables (PII)
  - Masquer les informations personnelles
  - Remplacer les informations personnelles
  - Chiffrer les informations personnelles
  - Réorganisation des lignes
- Étapes de la recette de structure des colonnes
  - Split
  - Fusionner
  - Créer
- Étapes de la recette de mise en forme

  - Précision décimale
  - Séparateur de milliers
  - Numéros abrégés
- Étapes de recette de structure de données
  - Nest-Unnest
  - Pivot
  - Groupe
  - Joindre
  - Union
- Étapes de la recette de science des données
  - Texte
  - Échelle
  - Mappage
  - Codage
- Fonctions
  - Fonctions mathématiques
  - Fonctions d’agrégation
  - Fonctions de texte
  - Fonctions de date et d’heure
  - Fonctions de fenêtrage
  - Fonctions Web
  - Autres fonctions

Pour plus d'informations sur la manière dont ces étapes et fonctions de recette sont utilisées dans une recette (y compris l'utilisation d'expressions de condition), consultezDéfinition de la structure d'une recette.

Les sections suivantes décrivent les étapes et les fonctions de la recette, organisées en fonction de leur fonction.

Rubriques

Étapes de base de la recette des colonnes

Utilisez ces actions de recette de colonne de base pour effectuer des transformations simples sur vos données.

Rubriques

- RENAME
- SORT
- TO_BOOLEAN_COLUMN
- TO_DOUBLE_COLUMN
- TO_NUMBER_COLUMN
- TO_STRING_COLUMN

MODIFIER_TYPE DE DONNÉES

Modifie le type de données d'une colonne existante.

Si la valeur d'une colonne ne peut pas être convertie dans le nouveau type, elle sera remplacée par NULL. Cela peut se produire lorsqu'une colonne de chaîne est convertie en colonne de nombres entiers. Par exemple, la chaîne « 123 » deviendra un entier 123, mais la chaîne « ABC » ne peut pas devenir un nombre, elle sera donc remplacée par une valeur NULL.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Nouveau type de colonne. Les types de données suivants sont pris en charge :
  - octet : nombres entiers signés de 1 octet. La plage de nombres va de -128 à 127.
  - short : nombres entiers signés de 2 octets. La plage de nombres va de -32768 à 32767.
  - int : nombres entiers signés sur 4 octets. La plage de nombres va de -2147483648 à 2147483647.
  - long : nombres entiers signés de 8 octets. La plage de nombres va de -9223372036854775808 à 9223372036854775807.
  - float : nombres à virgule flottante à précision unique de 4 octets.
  - double : nombres à virgule flottante à double précision de 8 octets.
  - décimal : nombres décimaux signés comportant jusqu'à 38 chiffres au total et 18 chiffres après la virgule décimale.
  - chaîne : valeurs de chaîne de caractères.
  - booléen : le type booléen a l'une des deux valeurs possibles : « vrai » et « faux » ou « oui » et « non ».

- horodatage : valeurs comprenant les champs année, mois, jour, heure, minute et seconde.
- date : valeurs comprenant les champs année, mois et jour.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
 }
```

DELETE

Supprime une colonne de l'ensemble de données.

Parameters

- `sourceColumn` : nom d’une colonne existante.

Example Exemple

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

DUPLIQUER

Crée une nouvelle colonne portant un nom différent, mais contenant toutes les mêmes données. Les anciennes et les nouvelles colonnes sont conservées dans le jeu de données.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn`— Nom de la colonne dupliquée.

Example Exemple

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

JSON_TO_STRUCTS

Convertit une chaîne JSON en structures typées statiquement. Lors de la conversion, il détecte le schéma de chaque objet JSON et les fusionne afin d'obtenir le schéma le plus générique pour représenter l'intégralité de la chaîne JSON. Le paramètre « UnnestLevel » indique le nombre de niveaux d'objets JSON à convertir en structures.

Parameters

- `sourceColumns`— Liste des colonnes sources.
- `regexColumnSelector` –Expression régulière pour sélectionner les colonnes.
- `removeSourceColumn`— Valeur booléenne. `true`Si c'est le cas, supprimez la colonne source ; sinon, conservez-la.
- `unnestLevel`— Le nombre de niveaux à dénicher.
- `conditionExpressions`— Expressions conditionnelles.

Example Exemple

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

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

DÉPLACER_APRÈS

Déplace une colonne à la position immédiatement après une autre colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn`— Le nom d'une autre colonne. La colonne spécifiée par `sourceColumn` sera déplacée immédiatement après la colonne spécifiée par`targetColumn`.

Example Exemple

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

DÉPLACER_AVANT

Déplace une colonne à la position située juste avant une autre colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn`— Le nom d'une autre colonne. La colonne spécifiée par `sourceColumn` sera déplacée immédiatement après la colonne spécifiée par`targetColumn`.

Example Exemple

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

DÉPLACER VERS LA FIN

Déplace une colonne vers la position finale (dernière colonne) du jeu de données.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

DÉPLACER_VERS_INDEX

Déplace une colonne vers une position spécifiée par un nombre.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetIndex`— La nouvelle position de la colonne. Les positions commencent par 0. Ainsi, par exemple, cela 1 fait référence à la deuxième colonne, 2 à la troisième colonne, etc.

Example Exemple

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

PASSER AU POINT DE DÉPART

Déplace une colonne vers la position de départ (première colonne) du jeu de données.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

RENAME

Crée une nouvelle colonne portant un nom différent, mais contenant toutes les mêmes données. L'ancienne colonne est ensuite supprimée de l'ensemble de données.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `targetColumn`— Nouveau nom pour la colonne.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

Trie les données dans une ou plusieurs colonnes d'un ensemble de données par ordre croissant, décroissant ou personnalisé.

Parameters

- `expressions`— Chaîne contenant une ou plusieurs JSON-encoded chaînes représentant des expressions de tri.
  - `sourceColumn`— Chaîne contenant le nom d'une colonne existante.
  - `ordering`— La commande peut être ascendante ou descendante.
  - `nullsOrdering`— L'ordre des valeurs nulles peut être NULLS_TOP ou NULLS_BOTTOM pour placer les valeurs nulles ou manquantes au début ou au bas de la colonne.
  - `customOrder`— Liste de chaînes qui définit un ordre personnalisé pour le tri des chaînes. Par défaut, les chaînes sont triées par ordre alphabétique.
  - `isCustomOrderCaseSensitive` : booléen. La valeur par défaut est `false`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SORT",
```

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

Example Exemple d'ordre de tri personnalisé

Dans l'exemple suivant, la chaîne d'expression CustomOrder a le format d'une liste d'objets. Chaque objet décrit une expression de tri pour une colonne.

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

TO_BOOLEAN_COLUMN

Modifie le type de données d'une colonne existante en BOOLEAN.

> **Note**
>
> Nous recommandons d'utiliser l'action de recette CHANGE_DATA_TYPE plutôt que TO_BOOLEAN_COLUMN.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `columnDataType`— Une valeur qui doit être`boolean`.

Example Exemple

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

TO_DOUBLE_COLUMN

Modifie le type de données d'une colonne existante en DOUBLE.

> **Note**
>
> Nous recommandons d'utiliser l'action de recette CHANGE_DATA_TYPE plutôt que TO_DOUBLE_COLUMN.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Une valeur qui doit êtrenumber.

Example Exemple

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

```
    }
}
```

TO_NUMBER_COLUMN

Modifie le type de données d'une colonne existante en NUMBER.

> **Note**
>
> Nous recommandons d'utiliser l'action de recette CHANGE_DATA_TYPE plutôt que TO_NUMBER_COLUMN.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Une valeur qui doit êtrenumber.

Example Exemple

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

TO_STRING_COLUMN

Remplace le type de données d'une colonne existante en STRING.

> **Note**
>
> Nous recommandons d'utiliser l'action de recette CHANGE_DATA_TYPE plutôt que TO_STRING_COLUMN.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Une valeur qui doit êtrestring.

Example Exemple

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

Étapes de la recette de nettoyage des données

Utilisez ces étapes de la recette de nettoyage des données pour effectuer des transformations simples sur des données existantes.

Rubriques

- MODÈLE_EXTRAIT
- VALEUR_D'EXTRACTION
- SUPPRIMER_COMBINÉ
- REMPLACER_ENTRE_DÉLIMITEURS
- REMPLACER_ENTRE_POSITIONS
- REMPLACER_TEXTE

MAJUSCULE_CASE

Modifie chaque chaîne d'une colonne pour mettre chaque mot en majuscule. En majuscules, la première lettre de chaque mot est en majuscule et le reste du mot est transformé en minuscules. Un exemple est : Le renard brun rapide a sauté par-dessus la clôture.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

DATE_FORMAT

Renvoie une colonne dans laquelle une chaîne de date est convertie en valeur formatée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetDateFormat`— L'un des formats de date suivants :

- mm/dd/yyyy
- mm-dd-yyyy
- dd month yyyy
- month yyyy
- dd month

Example Exemple

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

MINUSCULE

Fait passer chaque chaîne d'une colonne en minuscules, par exemple : le rapide renard brun a sauté par-dessus la clôture

Parameters

- sourceColumn : nom d’une colonne existante.

Example Exemple

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

MAJUSCULE_CASE

Remplace chaque chaîne d'une colonne en majuscules, par exemple : LE RENARD BRUN A SAUTÉ PAR-DESSUS LA CLÔTURE

Parameters

- `sourceColumn` : nom d’une colonne existante.

Example Exemple

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

PHRASE_CASE

Modifie chaque chaîne d'une colonne en majuscules. Dans le cas d'une phrase, la première lettre de chaque phrase est en majuscule et le reste de la phrase est transformé en minuscules. Un exemple est : le renard brun rapide. J'ai sauté par-dessus. La clôture

Parameters

- `sourceColumn` : nom d’une colonne existante.

Example Exemple

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

```
    }
}
```

AJOUTER_DOUBLE_QUOTES

Met les caractères entre guillemets doubles dans une colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

AJOUTER_PRÉFIXE

Ajoute un ou plusieurs caractères en les concaténant sous forme de préfixe au début d'une colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Le ou les caractères à placer au début des valeurs de colonne.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ADD_PREFIX",
        "Parameters": {
            "pattern": "aaa",
```

```
            "sourceColumn": "info_url"
        }
    }
}
```

AJOUTER_UN_CITATIONS

Met les caractères entre guillemets simples dans une colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

AJOUTER_SUFFIXE

Ajoute un caractère supplémentaire en les concaténant sous forme de suffixe à la fin d'une colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Le ou les caractères à placer à la fin de la colonne.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ADD_SUFFIX",
```

```
        "Parameters": {
            "pattern": "bbb",
            "sourceColumn": "info_url"
        }
    }
}
```

EXTRACTIVER_ENTRE_DÉLIMITEURS

Crée une nouvelle colonne, basée sur des délimiteurs, à partir des valeurs d'une colonne existante.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `startPattern`— Expression régulière indiquant le ou les caractères commençant par les valeurs délimitées.
- `endPattern`— Expression régulière indiquant le ou les caractères séparateurs qui terminent les valeurs délimitées.

Example Exemple

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

EXTRAYER_ENTRE_POSITIONS

Crée une nouvelle colonne, basée sur la position des caractères, à partir des valeurs d'une colonne existante.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `startPosition`— Position du personnage à laquelle effectuer l'extraction.
- `endPosition`— Position du caractère à laquelle terminer l'extrait.

Example Exemple

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

MODÈLE_EXTRAIT

Crée une nouvelle colonne, basée sur une expression régulière, à partir des valeurs d'une colonne existante.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `pattern`— Expression régulière qui indique le ou les caractères à extraire et à partir desquels créer la nouvelle colonne.

Example Exemple

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

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

VALEUR_D'EXTRACTION

Crée une nouvelle colonne avec une valeur extraite d'un chemin spécifié par l'utilisateur. Si la colonne source est de type Map, Array ou Struct, chaque champ du chemin doit être évité à l'aide de backticks (par exemple, « name »).

Parameters

- `targetColumn`— Nom de la colonne cible.
- `sourceColumn`— Nom de la colonne source à partir de laquelle la valeur doit être extraite.
- `path`— Le chemin d'accès à la clé spécifique que l'utilisateur souhaite extraire. Si la colonne source est de type Map, Array ou Struct, chaque champ du chemin doit être évité à l'aide de backticks (par exemple, « name »).

  Prenons l'exemple suivant d'informations utilisateur :

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

  Voici des exemples de chemins que vous pouvez fournir, en fonction du type de colonne source :

- Si la colonne source est du type map, le chemin pour extraire le numéro de téléphone fixe est le suivant :

`user`.`phoneNumber`.`home`

- Si la colonne source est de type tableau, le chemin pour extraire la deuxième valeur de « citoyenneté » est le suivant :

  `user`.`citizenship`[1]

- Si la colonne source est de type struct, le chemin d'extraction du code postal est le suivant :

  `user`.`address`.`zipcode`

Example Exemple

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

SUPPRIMER_COMBINÉ

Supprime un ou plusieurs caractères d'une colonne, en fonction des informations spécifiées par l'utilisateur.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `collapseConsecutiveWhitespace`— Si`true`, remplace deux espaces blancs ou plus par exactement un caractère d'espace blanc.
- `removeAllPunctuation`— Si`true`, supprime tous les caractères suivants : `. ! , ?`
- `removeAllQuotes`— Si`true`, supprime tous les guillemets simples et doubles.
- `removeAllWhitespace`— Si`true`, supprime tous les espaces blancs.
- `customCharacters`— Un ou plusieurs personnages sur lesquels il est possible d'agir.

- `customValue`— Une valeur sur laquelle il est possible d'agir.
- `removeCustomCharacters`— Si`true`, supprime tous les caractères spécifiés par `customCharacters` paramètre.
- `removeCustomValue`— Si`true`, supprime tous les caractères spécifiés par `customValue` paramètre.
- `punctuationally`— Si`true`, supprime les caractères suivants s'ils apparaissent au début ou à la fin de la valeur : `. ! , ?`
- `antidisestablishmentarianism`— Si`true`, supprime les guillemets simples et les guillemets doubles au début et à la fin de la valeur.
- `removeLeadingAndTrailingWhitespace`— Si`true`, supprime tous les espaces blancs situés au début et à la fin de la valeur.
- `removeLetters`— Si`true`, supprime tous les caractères alphabétiques majuscules et minuscules (de `A` jusqu'à`Z`). `a` `z`
- `removeNumbers`— Si`true`, supprime tous les caractères numériques (`0`jusqu'à`9`).
- `removeSpecialCharacters`— Si`true`, supprime tous les caractères suivants : `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`

Example Exemples

```
{
    "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"
        }
```

```
    }
}
```

REMPLACER_ENTRE_DÉLIMITEURS

Remplace les caractères situés entre deux délimiteurs par du texte défini par l'utilisateur.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `startPattern`— Caractère ou caractères ou expression régulière indiquant le point de départ de la substitution.
- `endPattern`— Caractère ou caractères ou expression régulière indiquant la fin de la substitution.
- `value`— Le ou les caractères de remplacement à substituer.

Example Exemple

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

REMPLACER_ENTRE_POSITIONS

Remplace les caractères situés entre deux positions par du texte défini par l'utilisateur.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `startPosition`— Nombre indiquant à quelle position de caractère de la chaîne la substitution doit commencer.

- `endPosition`— Un nombre indiquant à quelle position de caractère de la chaîne la substitution doit se terminer.
- `value`— Le ou les caractères de remplacement à substituer.

Example Exemple

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

REMPLACER_TEXTE

Remplace une séquence de caractères spécifiée par une autre.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Caractère ou caractères ou expression régulière, indiquant quels caractères doivent être remplacés dans la colonne source.
- `value`— Le ou les caractères de remplacement à substituer.

Example Exemples

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

```
        }
    }
}
```

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

Étapes de la recette de qualité des données

Utilisez ces étapes de recette de qualité des données pour renseigner les valeurs manquantes, supprimer les données non valides ou supprimer les doublons.

Rubriques

- FILTRE_TYPE DE DONNÉES AVANCÉ
- DRAPEAU DE TYPE DE DONNÉES AVANCÉ
- SUPPRIMER_DUPLIQUER_LIGNES
- EXTRACT_ADVANCED_DATATYPE_DETAILS
- REMPLISSEZ_AVEC_MOYENNE
- REMPLISSEZ_AVEC_CUSTOM
- REMPLIR_AVEC_VIDE
- REMPLISSEZ_AVEC_DERNIER_VALIDE
- REMPLISSEZ_AVEC_MÉDIANE
- REMPLISSEUR_AVEC_MODE
- REMPLISSEZ_AVEC_MOST_FREQUENT
- REMPLIR_AVEC_NULL
- REMPLIR_AVEC_SOMME
- FLAG_DUPLICATE_ROWS

- LE DRAPEAU SE DUPLIQUE DANS UNE COLONNE
- GET_ADVANCED_DATATYPE
- SUPPRIMER_DUPLICATES
- SUPPRIMER_INVALIDE
- SUPPRIMER_MANQUANT
- REMPLACER_PAR_MOYEN
- REMPLACER_PAR_PERSONNALISÉ
- REMPLACER_PAR_VIDE
- REMPLACER_AVEC_DERNIER_VALID
- REMPLACER_PAR_MÉDIAN
- REMPLACER_PAR_MODE
- REMPLACER_PAR_PLUS_FRÉQUENT
- REMPLACER_PAR_NUL
- REMPLACER_PAR_ROLLING_AVERAGE
- REMPLACER_PAR_ROLLING_SUM
- REMPLACER_PAR_SOMME

FILTRE_TYPE DE DONNÉES AVANCÉ

Filtre la colonne source actuelle en fonction de la détection avancée des types de données. Par exemple, étant donné une colonne DataBrew identifiée comme contenant des codes postaux, cette transformation peut filtrer la colonne en fonction du fuseau horaire. Les détails que vous pouvez extraire dépendent du modèle détecté, comme décrit dans les notes ci-dessous.

Parameters

- `sourceColumn`— Nom d'une colonne source sous forme de chaîne.
- `pattern`— Le motif à extraire.
- `advancedDataType`— Il peut s'agir du téléphone, du code postal, de la date, de l'heure, de l'État, de la carte de crédit, de l'URL, du courrier électronique, du SSN ou du sexe.
- `filter values`— Liste des valeurs de chaîne sur lesquelles l'utilisateur souhaite filtrer la colonne.
- `strategy`— KEEP_ROWS ou DISCARD_ROWS ou CLEAR_FILTERS ou CLEAR_OTHERS.

- `clearWithEmpty`— Booléen `true` ou`false`, pour effacer des lignes `empty` au lieu de. `null`

Remarques

- Si la valeur avancée DataType est Phone, le modèle peut être AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Si la valeur avancée DataType est le code postal, le modèle peut être TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.
- Si la valeur avancée DataType est Date/Heure, le modèle peut être DAY, MONTH_NAME, WEEK, QUARTER ou YEAR.
- Si advanced DataType est défini sur State, le modèle peut être TIME_ZONE.
- Si la DataType valeur avancée est Carte de crédit, le modèle peut être LONGUEUR ou RÉSEAU.
- Si advanced DataType est URL, le modèle peut être PROTOCOL, TLD ou DOMAIN.

Example Exemple

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

DRAPEAU DE TYPE DE DONNÉES AVANCÉ

Crée une nouvelle colonne d'indicateurs basée sur les valeurs de la colonne source actuelle. Par exemple, étant donné une colonne source contenant des codes postaux, cette transformation peut être utilisée pour marquer des valeurs en `false` fonction `true` ou en fonction d'un fuseau horaire particulier. Les détails que vous pouvez extraire dépendent du modèle détecté, comme décrit dans les notes ci-dessous.

Parameters

- `sourceColumn`— Nom d'une colonne source sous forme de chaîne.
- `pattern`— Le motif à extraire.
- `targetColumn`— Nom de la colonne cible.
- `advancedDataType`— Il peut s'agir du téléphone, du code postal, de la date, de l'heure, de l'État, de la carte de crédit, de l'URL, du courrier électronique, du SSN ou du sexe.
- `filter values`— Liste des valeurs de chaîne sur lesquelles l'utilisateur souhaite filtrer la colonne.
- `trueString`— `true` Valeur de la colonne cible.
- `falseString`— `false` Valeur de la colonne cible.

Remarques

- Si la valeur avancée DataType est Phone, le modèle peut être AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Si la valeur avancée DataType est le code postal, le modèle peut être TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.
- Si la valeur avancée DataType est Date/Heure, le modèle peut être DAY, MONTH_NAME, WEEK, QUARTER ou YEAR.
- Si advanced DataType est défini sur State, le modèle peut être TIME_ZONE.
- Si la DataType valeur avancée est Carte de crédit, le modèle peut être LONGUEUR ou RÉSEAU.
- Si advanced DataType est URL, le modèle peut être PROTOCOL, TLD ou DOMAIN.

Example Exemple

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

```
            "trueString": "trueValue",
            "falseString": "falseValue"
        }
    }
}
```

SUPPRIMER_DUPLIQUER_LIGNES

Supprime toute ligne correspondant exactement à une ligne précédente du jeu de données. L'occurrence initiale n'est pas supprimée car elle ne correspond pas à une ligne précédente.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRACT_ADVANCED_DATATYPE_DETAILS

Extrait les détails du type de données avancé. Les détails que vous pouvez extraire dépendent du modèle détecté, comme décrit dans les notes ci-dessous.

Parameters

- `sourceColumn`— Nom d'une colonne source sous forme de chaîne.
- `pattern`— Le motif à extraire.
- `targetColumn`— Nom de la colonne cible.
- `advancedDataType`— Il peut s'agir du téléphone, du code postal, de la date, de l'heure, de l'État, de la carte de crédit, de l'URL, du courrier électronique, du SSN ou du sexe.

Remarques

- Si la valeur avancée DataType est Phone, le modèle peut être AREA_CODE, TIME_ZONE ou COUNTRY_CODE.
- Si la valeur avancée DataType est le code postal, le modèle peut être TIME_ZONE, COUNTRY, STATE, CITY, TYPE ou REGION.

- Si la valeur avancée DataType est Date/Heure, le modèle peut être DAY, MONTH_NAME, WEEK, QUARTER ou YEAR.
- Si advanced DataType est défini sur State, le modèle peut être TIME_ZONE.
- Si la DataType valeur avancée est Carte de crédit, le modèle peut être LONGUEUR ou RÉSEAU.
- Si advanced DataType est URL, le modèle peut être PROTOCOL, TLD ou DOMAIN.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

REMPLISSEZ_AVEC_MOYENNE

Renvoie une colonne dans laquelle les données manquantes sont remplacées par la moyenne de toutes les valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
```

```
}
```

REMPLISSEZ_AVEC_CUSTOM

Renvoie une colonne dont les données manquantes sont remplacées par une valeur spécifique.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit être `datenumber`,`boolean`, `unsupportedstring`, `outimestamp`.
- `value`— La valeur personnalisée à renseigner. Le type de données doit correspondre à la valeur que vous avez choisie`columnDataType`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

REMPLIR_AVEC_VIDE

Renvoie une colonne contenant des données manquantes remplacées par une chaîne vide.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
```

```
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
}
```

REMPLISSEZ_AVEC_DERNIER_VALIDE

Renvoie une colonne dont les données manquantes sont remplacées par la valeur valide la plus récente pour cette colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit être `datenumber`,`boolean`, `unsupportedstring`, ou`timestamp`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

REMPLISSEZ_AVEC_MÉDIANE

Renvoie une colonne dont les données manquantes sont remplacées par la médiane de toutes les valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

REMPLISSEUR_AVEC_MODE

Renvoie une colonne dont les données manquantes sont remplacées par le mode de toutes les valeurs.

Vous pouvez également définir une logique de disjoncteur de lien, dans laquelle certaines valeurs sont identiques. Par exemple, considérez les valeurs suivantes :

`1 2 2 3 3 4`

A `modeType` de `MINIMUM` provoque `FILL_WITH_MODE` le renvoi de 2 comme valeur du mode. Si tel `modeType` est le cas`MAXIMUM`, le mode est 3. Pour `AVERAGE` le mode est 2,5.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `modeType` : comment résoudre les valeurs à égalité dans les données. Cette valeur doit être `MINIMUM` `NONEAVERAGE`, ou`MAXIMUM`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
```

```
    }
}
```

REMPLISSEZ_AVEC_MOST_FREQUENT

Renvoie une colonne dont les données manquantes sont remplacées par la valeur la plus fréquente.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
    }
}
```

REMPLIR_AVEC_NULL

Renvoie une colonne dont les valeurs de données sont remplacées par des valeurs nulles.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
    }
```

```
}
```

REMPLIR_AVEC_SOMME

Renvoie une colonne dont les données manquantes sont remplacées par la somme de toutes les valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FLAG_DUPLICATE_ROWS

Renvoie une nouvelle colonne avec une valeur spécifiée dans chaque ligne qui indique si cette ligne correspond exactement à une ligne précédente de l'ensemble de données. Lorsque des correspondances sont trouvées, elles sont signalées comme des doublons. L'occurrence initiale n'est pas signalée, car elle ne correspond pas à une ligne précédente.

Parameters

- `trueString` : valeur à insérer si la ligne correspond à une ligne précédente.
- `falseString` : valeur à insérer si la ligne est unique.
- `targetColumn` : nom de la nouvelle colonne insérée dans le jeu de données.

Example Exemple

```
{
```

```
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

LE DRAPEAU SE DUPLIQUE DANS UNE COLONNE

Renvoie une nouvelle colonne avec une valeur spécifiée dans chaque ligne qui indique si la valeur de la colonne source de la ligne correspond à une valeur d'une ligne précédente de la colonne source. Lorsque des correspondances sont trouvées, elles sont signalées comme des doublons. L'occurrence initiale n'est pas signalée, car elle ne correspond pas à une ligne précédente.

Parameters

- `sourceColumn` : nom de la colonne source.
- `targetColumn` : nom de la colonne cible.
- `trueString` : chaîne à insérer dans la colonne cible lorsqu'une valeur de colonne source duplique une valeur antérieure dans cette colonne.
- `falseString` : chaîne à insérer dans la colonne cible lorsqu'une valeur de colonne source est différente des valeurs précédentes dans cette colonne.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
}
```

GET_ADVANCED_DATATYPE

Étant donné une colonne de chaîne, identifie le type de données avancé de la colonne, le cas échéant.

Parameters

- `columnName`— Nom de la colonne de chaîne.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

SUPPRIMER_DUPLICATES

Supprime une ligne entière si une valeur dupliquée est détectée dans une colonne source sélectionnée.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

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

SUPPRIMER_INVALIDE

Supprime une ligne entière si une valeur non valide est détectée dans une colonne de cette ligne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une `string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
}
```

SUPPRIMER_MANQUANT

Renvoie uniquement les lignes dans lesquelles aucune donnée ne manque à une colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REMOVE_MISSING",
```

```
        "Parameters": {
            "sourceColumn": "last_name"
        }
    }
}
```

REMPLACER_PAR_MOYEN

Remplace chaque valeur non valide d'une colonne par la moyenne de toutes les autres valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit être`number`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "age"
        }
    }
}
```

REMPLACER_PAR_PERSONNALISÉ

Remplacez les entités détectées par une valeur personnalisée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `sourceColumns`— Liste des noms de colonnes existants.
- `columnDataType`— Type de données de la colonne.
- `value`— La valeur personnalisée à utiliser pour remplacer les valeurs non valides.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une

`string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

> **Note**
>
> Utilisez l'un `sourceColumn` ou l'autre`sourceColumns`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

REMPLACER_PAR_VIDE

Remplace chaque valeur non valide d'une colonne par une valeur vide.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une `string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

REMPLACER_AVEC_DERNIER_VALID

Remplace chaque valeur non valide d'une colonne par la dernière valeur valide.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une `string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
    }
}
```

REMPLACER_PAR_MÉDIAN

Remplace chaque valeur non valide d'une colonne par la médiane de toutes les autres valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit être`number`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

REMPLACER_PAR_MODE

Remplace chaque valeur non valide d'une colonne par le mode de toutes les autres valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit être`number`.
- `modeType` : comment résoudre les valeurs à égalité dans les données. Cette valeur doit être `MINIMUM` `NONEAVERAGE`, ou`MAXIMUM`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
```

```
    }
}
```

REMPLACER_PAR_PLUS_FRÉQUENT

Remplace chaque valeur non valide d'une colonne par la valeur de colonne la plus fréquente.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une `string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "wind_direction"
        }
    }
}
```

REMPLACER_PAR_NUL

Remplace chaque valeur non valide d'une colonne par une valeur nulle.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne.
- `advancedDataType`— Types de données spéciaux détectés DataBrew dans une colonne contenant le type de données`string`. Les types qui DataBrew peuvent être détectés dans une

`string` colonne incluent le SSN, l'e-mail, le numéro de téléphone, le sexe, la carte de crédit, l'URL, l'adresse IP DateTime, la devise ZipCode, le pays, la région, l'État et la ville.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

REMPLACER_PAR_ROLLING_AVERAGE

Remplace chaque valeur d'une colonne par la moyenne mobile d'une « fenêtre » de lignes précédente.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit êtrenumber`.`
- `period`- — La taille de la fenêtre. Par exemple, si la `period` valeur est 10, la moyenne mobile est calculée en utilisant les 10 lignes précédentes.

Example Exemple

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
```

```
        }
    }
}
```

REMPLACER_PAR_ROLLING_SUM

Remplace chaque valeur d'une colonne par la somme mobile d'une « fenêtre » de lignes précédente.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit êtrenumber.
- `period`- — La taille de la fenêtre. Par exemple, si la `period` valeur est 10, la somme mobile est calculée en utilisant les 10 lignes précédentes.

Example Exemple

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REMPLACER_PAR_SOMME

Remplace chaque valeur non valide d'une colonne par la somme de toutes les autres valeurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `columnDataType`— Type de données de la colonne. Ce type doit êtrenumber.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

Étapes de la recette des informations personnellement identifiables (PII)

Utilisez ces étapes de recette pour effectuer des transformations sur les informations personnelles identifiables (PII) dans un ensemble de données.

> **Note**
>
> Outre les étapes de recette décrites dans cette section, il existe des étapes de DataBrew recette non conçues spécifiquement pour les informations personnelles que vous pouvez utiliser pour gérer les informations personnelles. Par exempleDELETE, une étape de recette de colonne de base qui supprime une colonne.

Rubriques

- HACHAGE_CRYPTOGRAPHIQUE
- DÉCRYPTER
- DÉCHIFFRE_DÉTERMINISTE
- CHIFFRE_DÉTERMINISTE
- CRYPTER
- MASQUE_PERSONNALISÉ
- MASQUE_DATE

- MASK_DELIMITER
- MASK_RANGE
- REMPLACER_PAR_RANDOM_BETWEEN
- REMPLACER_PAR_DATE_RANDOM_BETWEEN
- SHUFFLE_ROWS

HACHAGE_CRYPTOGRAPHIQUE

Applique un algorithme aux valeurs de hachage de la colonne.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `secretId` : ARN de la clé secrète Secrets Manager La clé utilisée dans l'algorithme de préfixe du code d'authentification des messages basé sur le hachage (HMAC) pour hacher les colonnes source, ou `databrew!default` est la sortie décodée en base64 pour la valeur de la clé secrète de Secrets Manager.
- `secretVersion` : facultatif. Par défaut, la version du secret la plus récente.
- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.
- `createSecretIfMissing` : valeur booléenne facultative. Si la valeur est true, il essaiera de créer le secret au nom de l'appelant.
- `algorithm` : l'algorithme utilisé pour hacher vos données. Valeurs d'énumération valides : MD5, SHA1, SHA256, SHA512, HMAC_MD5, HMAC_SHA1, HMAC_SHA256, HMAC_SHA512

  Chaque option fait référence à un algorithme de hachage différent. Les options avec le préfixe « HMAC » font référence à un algorithme de hachage à clé et nécessitent le paramètre. `secretId` Pour les options sans le préfixe « HMAC », le `secretId` paramètre n'est pas obligatoire.

  Si vous ne fournissez pas d'algorithme de hachage, le service prend par défaut « HMAC_SHA256 ».

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
```

```
   "entityTypeFilter": ["USA_ALL"]
}
```

Lorsqu'il travaille dans le cadre de l'expérience interactive, outre le rôle du projet, l'utilisateur de la console doit être autorisé à `secretsmanager:GetSecretValue` accéder au secret Secrets Manager fourni.

Exemple de politique :

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

Vous pouvez également choisir d'utiliser le secret DataBrew-created par défaut en le transmettant `databrew!default` comme SecreTid et `createSecretIfMissing` le paramètre comme true. Ceci n'est pas recommandé pour la production. Toute personne possédant le AwsGlueDataBrewFullAccessPolicyrôle peut utiliser le secret par défaut.

DÉCRYPTER

Vous pouvez utiliser la transformation DECRYPT pour déchiffrer l'intérieur de. DataBrew Vos données peuvent également être déchiffrées en dehors de l'extérieur à l' DataBrew aide du AWS SDK de chiffrement. Si l'ARN de la clé KMS fourni ne correspond pas à celui utilisé pour chiffrer la colonne, l'opération de déchiffrement échoue. Pour plus d'informations sur le SDK de AWS chiffrement, voir Qu'est-ce que le SDK de AWS chiffrement dans le guide du AWS Encryption SDK développeur.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `kmsKeyArn`— L'ARN de la AWS clé du service de gestion des clés à utiliser pour déchiffrer les colonnes source. Pour plus d'informations sur l'ARN clé, consultez la section Key ARN dans le guide du AWS Key Management Service développeur.

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
}
```

Lorsqu'il travaille dans le cadre de l'expérience interactive, outre le rôle du projet, l'utilisateur de la console doit être autorisé à utiliser `kms:GenerateDataKey` et `kms:Decrypt` à utiliser la clé KMS fournie.

Exemple de politique :

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"
        ]
    }
  ]
}
```

DÉCHIFFRE_DÉTERMINISTE

Déchiffre les données chiffrées avec DETERMINISTIC_ENCRYPT.

Cette transformation est interdite si l'identifiant secret et la version fournis ne correspondent pas à ceux utilisés pour chiffrer la colonne.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `secretId`— L'ARN de la clé secrète Secrets Manager à utiliser pour déchiffrer les colonnes sources.
- `secretVersion` : facultatif. Par défaut, la version du secret la plus récente.

Exemple

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

Lorsqu'il travaille dans l'expérience interactive, outre le rôle du projet, l'utilisateur de la console doit avoir l'autorisation d'accéder à secretsmanager : GetSecretValue sur le secret Secrets Manager fourni.

Exemple de politique :

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

CHIFFRE_DÉTERMINISTE

Chiffre la colonne à l'aide AES-GCM-SIV d'une clé de 256 bits. Les données chiffrées avec DETERMINISTIC_ENCRYPT ne peuvent être déchiffrées qu'à l'intérieur ou avec la transformation DETERMINISTIC_DECRYPT. DataBrew Cette transformation n'utilise AWS KMS pas le SDK de AWS chiffrement, mais utilise à la place la bibliothèque AWS LC github.

Peut chiffrer jusqu'à 400 Ko par cellule. Ne préserve pas le type de données lors du déchiffrement.

> **Note**
>
> Remarque : Il est déconseillé d'utiliser un secret pendant plus d'un an.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `secretId`— L'ARN de la clé secrète Secrets Manager à utiliser pour chiffrer les colonnes sources, ou databrew ! par défaut.
- `secretVersion` : facultatif. Par défaut, la version du secret la plus récente.
- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.
- `createSecretIfMissing` : valeur booléenne facultative. Si la valeur est true, il essaiera de créer le secret au nom de l'appelant.

Exemple

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123",
   "entityTypeFilter": ["USA_ALL"]
}
```

Lorsqu'il travaille dans le cadre de l'expérience interactive, outre le rôle du projet, l'utilisateur de la console doit être autorisé à `secretsmanager:GetSecretValue` accéder au secret Secrets Manager fourni.

Exemple de politique

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

CRYPTER

Chiffre les valeurs des colonnes source à l'aide du SDK de AWS chiffrement. La transformation DECRYPT peut être utilisée pour déchiffrer l'intérieur de. DataBrew Vous pouvez également déchiffrer les données en dehors de l' DataBrew utilisation du SDK de AWS chiffrement.

La transformation ENCRYPT peut chiffrer jusqu'à 128 MiB par cellule. Elle tentera de préserver le format lors du déchiffrement. Pour préserver le type de données, les métadonnées du type de données doivent être sérialisées à moins de 1 Ko. Dans le cas contraire, vous devez définir le paramètre `preserveDataType` sur false. Les métadonnées du type de données seront stockées en texte clair dans le contexte du chiffrement. Pour plus d'informations sur le contexte de chiffrement, voir Contexte de chiffrement dans le Guide du AWS Key Management Service développeur.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `kmsKeyArn`— L'ARN AWS clé du service de gestion des clés à utiliser pour chiffrer les colonnes source. Pour plus d'informations sur l'ARN clé, consultez la section Key ARN dans le guide du AWS Key Management Service développeur.

- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.
- `preserveDataType` : valeur booléenne facultative. La valeur par défaut est true (vrai). Si la valeur est false, le type de données ne sera pas stocké.

Dans l'exemple suivant, `entityTypeFilter` et `preserveDataType` sont facultatifs.

Exemple

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

Lorsque vous travaillez dans l'expérience interactive, outre le rôle du projet, l'utilisateur de la console doit être autorisé `kms:GenerateDataKey` à utiliser la AWS KMS clé fournie.

Exemple de politique :

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

MASQUE_PERSONNALISÉ

Masque les caractères qui correspondent à une valeur personnalisée fournie.

Parameters

- `sourceColumns`— Liste des noms de colonnes existants.
- `maskSymbol`— Symbole qui sera utilisé pour remplacer les caractères spécifiés.
- `regex`— Si c'est vrai, traite `customValue` comme un modèle de regex correspondant.
- `customValue`— Toutes les occurrences (ou correspondances régulières) de `customValue` seront masquées dans la chaîne.
- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.

Example Exemple

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
        "Parameters": {
            "sourceColumns": ["company"],
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
}
```

MASQUE_DATE

Masque les composants d'une date à l'aide d'un symbole de masque défini par l'utilisateur.

Parameters

- `sourceColumns`— Liste des noms de colonnes existants.
- `maskSymbol`— Symbole qui sera utilisé pour remplacer les caractères spécifiés.
- `redact`— Un tableau d'énumérations de composants de date à masquer. Valeurs d'énumération valides : ANNÉE, MOIS, JOUR, HEURE, MINUTE, SECONDE, MILLISECONDE.

- `locale`— Balise de langue IETF BCP 47 optionnelle. La valeur par défaut est `en` . Les paramètres régionaux à utiliser pour le formatage des dates.

Example Exemple

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

MASK_DELIMITER

Masque les caractères situés entre deux délimiteurs à l'aide d'un symbole de masquage défini par l'utilisateur.

Parameters

- `sourceColumns`— Liste des noms de colonnes existants.
- `maskSymbol`— Symbole qui sera utilisé pour remplacer les caractères spécifiés.
- `startDelimiter`— Un caractère indiquant le point de départ du masquage. L'omission de ce paramètre appliquera le masque à partir du début de la chaîne.
- `endDelimiter`— Un caractère indiquant la fin du masquage. L'omission de ce paramètre appliquera le masquage du StartDelimiter à la fin de la chaîne.
- `preserveDelimiters`— Si vrai, applique un masque aux délimiteurs.
- `alphabet`— Un ensemble de jeux de caractères à conserver lors du masquage. Valeurs d'énumération valides : SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.

Example Exemple

```
// 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

Masque les caractères situés entre deux positions à l'aide d'un symbole de masquage défini par l'utilisateur.

Parameters

- `sourceColumns`— Liste des noms de colonnes existants.
- `maskSymbol`— Symbole qui sera utilisé pour remplacer les caractères spécifiés.
- `start`— Un chiffre indiquant à quelle position de caractère le masquage doit commencer (indexé à 0, inclus). L'indexation négative est autorisée. L'omission de ce paramètre appliquera le masque depuis le début de la chaîne jusqu'à « stop ».
- `stop`— Un chiffre indiquant à quelle position de caractère le masquage doit prendre fin (indexé 0, exclusif). L'indexation négative est autorisée. L'omission de ce paramètre appliquera le masque du début à la fin de la chaîne.
- `alphabet`— Un tableau d'énumérations de jeux de caractères à conserver lors du masquage. Valeurs d'énumération valides : SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Tableau facultatif de types d'entités. Peut être utilisé pour chiffrer uniquement les informations personnelles détectées dans une colonne en texte libre.

Example Exemple

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

REMPLACER_PAR_RANDOM_BETWEEN

Remplace les valeurs par un nombre aléatoire.

Parameters

- `lowerBound`— Limite inférieure de la plage de nombres aléatoires.
- `sourceColumns`— Liste des noms de colonnes existants.
- `upperBound`— Limite supérieure de la plage de nombres aléatoires.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
    }
}
```

REMPLACER_PAR_DATE_RANDOM_BETWEEN

Remplace les valeurs par une date aléatoire.

Parameters

- `startDate`— Le début de la plage de dates à partir de laquelle une date aléatoire sera prise.
- `sourceColumns`— Liste des noms de colonnes existants.
- `endDate`— Fin de la plage de dates à partir de laquelle une date aléatoire sera prise.

Example Exemple

```
{
    "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

Répartit les valeurs d'une colonne donnée. Le remaniement peut se produire avec des valeurs groupées dans une colonne secondaire.

Parameters

- `sourceColumns` : tableau de colonnes existantes.
- `groupByColumns`— Un tableau de colonnes permettant de regrouper les colonnes source lors du brassage.

Example Exemple

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
}
```

Détection des valeurs aberrantes et gestion des étapes de la recette

Suivez ces étapes de recette pour traiter les valeurs aberrantes de vos données et effectuer des transformations avancées sur celles-ci.

Rubriques

FLAG_OUTLIERS

Renvoie une nouvelle colonne contenant une valeur personnalisable dans chaque ligne qui indique si la valeur de la colonne source est une valeur aberrante.

Parameters

- `sourceColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `targetColumn`— Spécifie le nom d'une nouvelle colonne dans laquelle les résultats de la stratégie d'évaluation des valeurs aberrantes doivent être insérés.
- `outlierStrategy`— Spécifie l'approche à utiliser pour détecter les valeurs aberrantes. Les valeurs valides sont les suivantes :
  - `Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type.
  - `MODIFIED_Z_SCORE`— Identifie une valeur comme une valeur aberrante lorsqu'elle s'écarte de la médiane d'une valeur supérieure au seuil d'écart absolu médian.
  - `IQR`— Identifie une valeur comme une valeur aberrante lorsqu'elle se situe au-delà du premier et du dernier quartile des données de colonne. L'intervalle interquartile (IQR) mesure où se situent les 50 % intermédiaires des points de données.

- `threshold`— Spécifie la valeur de seuil à utiliser lors de la détection des valeurs aberrantes. La `sourceColumn` valeur est identifiée comme une valeur aberrante si le score calculé avec le `outlierStrategy` dépasse ce nombre. La valeur par défaut est 3.
- `trueString`— Spécifie la valeur de chaîne à utiliser si une valeur aberrante est détectée. La valeur par défaut est « True ».
- `falseString`— Spécifie la valeur de chaîne à utiliser si aucune valeur aberrante n'est détectée. La valeur par défaut est « False ».

Les exemples suivants montrent la syntaxe d'une seule RecipeActionopération. Une recette contient au moins une RecipeStepopération et une étape de recette contient au moins une action de recette. Une action de recette exécute la transformation des données que vous spécifiez. Un groupe d'actions de recette s'exécute dans un ordre séquentiel pour créer le jeu de données final.

JSON

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe JSON. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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"
        }
    }
}
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

YAML

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe YAML. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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'
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

SUPPRIMER LES VALEURS ABERRANTES

Supprime les points de données considérés comme des valeurs aberrantes, en fonction des paramètres définis dans les paramètres.

Parameters

- `sourceColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `outlierStrategy`— Spécifie l'approche à utiliser pour détecter les valeurs aberrantes. Les valeurs valides sont les suivantes :
  - `Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type.
  - `MODIFIED_Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la médiane d'une valeur supérieure au seuil d'écart absolu médian.

  - `IQR`— Identifie une valeur comme une valeur aberrante lorsqu'elle se situe au-delà du premier et du dernier quartile des données de colonne. L'intervalle interquartile (IQR) mesure où se situent les 50 % intermédiaires des points de données.
- `threshold`— Spécifie la valeur de seuil à utiliser lors de la détection des valeurs aberrantes. La `sourceColumn` valeur est identifiée comme une valeur aberrante si le score calculé avec le `outlierStrategy` dépasse ce nombre. La valeur par défaut est 3.
- `removeType`— Spécifie le mode de suppression des données. Les valeurs valides sont `DELETE_ROWS` et `CLEAR`.
- `trimValue`— Spécifie s'il faut supprimer toutes les valeurs aberrantes ou certaines d'entre elles. Cette valeur booléenne par défaut est. `FALSE`
  - `FALSE`— Supprime toutes les valeurs aberrantes
  - `TRUE`— Supprime les valeurs aberrantes dont le classement se situe en dehors du seuil percentile spécifié dans et. `minValue` `maxValue`
- `minValue`— Indique la valeur percentile minimale pour la plage aberrante. La plage valide est comprise entre 0 et 100.
- `maxValue`— Indique la valeur percentile maximale pour la plage des valeurs aberrantes. La plage valide est comprise entre 0 et 100.

Les exemples suivants montrent la syntaxe d'une seule RecipeActionopération. Une recette contient au moins une RecipeStepopération et une étape de recette contient au moins une action de recette. Une action de recette exécute la transformation des données que vous spécifiez. Un groupe d'actions de recette s'exécute dans un ordre séquentiel pour créer le jeu de données final.

JSON

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe JSON. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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"
        }
    }
}
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

YAML

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe YAML. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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'
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

REPLACE_OUTLIERS

Met à jour les valeurs des points de données considérées comme des valeurs aberrantes, en fonction des paramètres définis dans les paramètres.

Parameters

- `sourceColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `outlierStrategy`— Spécifie l'approche à utiliser pour détecter les valeurs aberrantes. Les valeurs valides sont les suivantes :
  - `Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type.
  - `MODIFIED_Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la médiane d'une valeur supérieure au seuil d'écart absolu médian.
  - `IQR`— Identifie une valeur comme une valeur aberrante lorsqu'elle se situe au-delà du premier et du dernier quartile des données de colonne. L'intervalle interquartile (IQR) mesure où se situent les 50 % intermédiaires des points de données.
- `threshold`— Spécifie la valeur de seuil à utiliser lors de la détection des valeurs aberrantes. La `sourceColumn` valeur est identifiée comme une valeur aberrante si le score calculé avec le `outlierStrategy` dépasse ce nombre. La valeur par défaut est 3.
- `replaceType`— Spécifie la méthode à utiliser lors du remplacement des valeurs aberrantes. Les valeurs valides sont les suivantes :
  - `WINSORIZE_VALUES`— Spécifie l'utilisation des percentiles minimum et maximum pour plafonner les valeurs.
  - `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`— Indique le type de fonction modale à utiliser quand `replaceType` c'est le cas`REPLACE_WITH_MODE`. Les valeurs valides sont les suivantes : `MINMAX`, et`AVERAGE`.
- `minValue`— Indique la valeur percentile minimale pour la plage des valeurs aberrantes à appliquer en cas `trimValue` d'utilisation. La plage valide est comprise entre 0 et 100.

- `maxValue`— Indique la valeur percentile maximale pour la plage des valeurs aberrantes à appliquer lorsqu'elle `trimValue` est utilisée. La plage valide est comprise entre 0 et 100.
- `value`— Spécifie la valeur à insérer lors de l'utilisation`REPLACE_WITH_CUSTOM`.
- `trimValue`— Spécifie s'il faut supprimer toutes les valeurs aberrantes ou certaines d'entre elles. Cette valeur booléenne est définie sur `TRUE` when `replaceType` is `REPLACE_WITH_NULLREPLACE_WITH_MODE`, ou. `WINSORIZE_VALUES` Par défaut, c'est `FALSE` pour tous les autres.
  - `FALSE`— Supprime toutes les valeurs aberrantes
  - `TRUE`—Supprime les valeurs aberrantes dont le classement se situe en dehors du seuil de percentile spécifié dans et. `minValue` `maxValue`

Les exemples suivants montrent la syntaxe d'une seule RecipeActionopération. Une recette contient au moins une RecipeStepopération et une étape de recette contient au moins une action de recette. Une action de recette exécute la transformation des données que vous spécifiez. Un groupe d'actions de recette s'exécute dans un ordre séquentiel pour créer le jeu de données final.

JSON

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe JSON. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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"
        }
    }
}
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

YAML

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe YAML. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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'
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

RESCALE_OUTLIERS_WITH_Z_SCORE

Renvoie une nouvelle colonne avec une valeur aberrante redimensionnée dans chaque ligne, en fonction des paramètres définis. Cette action applique également une Z-score normalisation aux valeurs de données d'échelle linéaire pour qu'elles aient une moyenne (μ) de 0 et un écart type (σ) de 1. Nous recommandons cette action pour gérer les valeurs aberrantes.

Parameters

- `sourceColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `targetColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.

- `outlierStrategy`— Spécifie l'approche à utiliser pour détecter les valeurs aberrantes. Les valeurs valides sont les suivantes :
  - `Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type.
  - `MODIFIED_Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la médiane d'une valeur supérieure au seuil d'écart absolu médian.
  - `IQR`— Identifie une valeur comme une valeur aberrante lorsqu'elle se situe au-delà du premier et du dernier quartile des données de colonne. L'intervalle interquartile (IQR) mesure où se situent les 50 % intermédiaires des points de données.
- `threshold`— La valeur seuil à utiliser lors de la détection des valeurs aberrantes. La `sourceColumn` valeur est identifiée comme une valeur aberrante si le score calculé avec le `outlierStrategy` dépasse ce nombre. La valeur par défaut est 3.

Les exemples suivants montrent la syntaxe d'une seule RecipeActionopération. Une recette contient au moins une RecipeStepopération et une étape de recette contient au moins une action de recette. Une action de recette exécute la transformation des données que vous spécifiez. Un groupe d'actions de recette s'exécute dans un ordre séquentiel pour créer le jeu de données final.

JSON

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple d'opération DataBrew Recipe, en utilisant la syntaxe JSON. Pour des exemples de syntaxe présentant une liste d'actions de recette, voirDéfinition de la structure d'une recette.

Example Exemple 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"
        }
    }
}
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

YAML

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple d'opération DataBrew Recipe, en utilisant la syntaxe YAML. Pour des exemples de syntaxe présentant une liste d'actions de recette, voirDéfinition de la structure d'une recette.

Example Exemple en YAML

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: Z_SCORE
    threshold: '3'
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

RESCALE_OUTLIERS_WITH_SKEW

Renvoie une nouvelle colonne avec une valeur aberrante redimensionnée dans chaque ligne, en fonction des paramètres définis. Cette action permet de réduire l'asymétrie de distribution en appliquant le log ou la transformation racine spécifiée. Nous recommandons cette action pour traiter les données asymétriques.

Parameters

- `sourceColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `targetColumn`— Spécifie le nom d'une colonne numérique existante susceptible de contenir des valeurs aberrantes.
- `outlierStrategy`— Spécifie l'approche à utiliser pour détecter les valeurs aberrantes. Les valeurs valides sont les suivantes :

  - `Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la moyenne d'un écart supérieur au seuil d'écart type.
  - `MODIFIED_Z_SCORE`— Identifie une valeur comme valeur aberrante lorsqu'elle s'écarte de la médiane d'une valeur supérieure au seuil d'écart absolu médian.
  - `IQR`— Identifie une valeur comme une valeur aberrante lorsqu'elle se situe au-delà du premier et du dernier quartile des données de colonne. L'intervalle interquartile (IQR) mesure où se situent les 50 % intermédiaires des points de données.
- `threshold`— Spécifie la valeur de seuil à utiliser lors de la détection des valeurs aberrantes. La `sourceColumn` valeur est identifiée comme une valeur aberrante si le score calculé avec le `outlierStrategy` dépasse ce nombre. La valeur par défaut est 3.
- `skewFunction`— Spécifie la méthode à utiliser lors du remplacement des valeurs aberrantes. Les valeurs valides sont les suivantes :
  - LOG — Applique une transformation forte pour réduire les biais positifs et négatifs. Il s'agit d'un logarithme naturel (2,718281828).
  - ROOT (with`value = 3`) — Applique une transformation assez forte pour réduire les biais positifs et négatifs. (Racine cubique)
  - ROOT (with`value = 2`) — Applique une transformation modérée pour réduire uniquement l'inclinaison positive. (Racine carrée)
  - SQUARE — Applique une transformation modérée pour réduire l'inclinaison négative. (Carré)
  - Transformation personnalisée — Applique la transformation spécifiée `LOG` ou la `ROOT` transformation à l'aide du numéro personnalisé fourni dans le `value` paramètre.
- `value`— Spécifie la valeur à utiliser pour la transformation personnalisée. S'il s'`skewFunction`agit de LOG, cette valeur représente la base du journal. S'il s'`skewFunction`agit de ROOT, cette valeur représente la puissance de la racine.

Les exemples suivants montrent la syntaxe d'une seule RecipeActionopération. Une recette contient au moins une RecipeStepopération et une étape de recette contient au moins une action de recette. Une action de recette exécute la transformation des données que vous spécifiez. Un groupe d'actions de recette s'exécute dans un ordre séquentiel pour créer le jeu de données final.

JSON

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe JSON. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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"
        }
    }
}
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

YAML

Voici un exemple `RecipeAction` à utiliser en tant que membre d'un `RecipeStep` exemple de DataBrew recette, en utilisant la syntaxe YAML. Pour des exemples de syntaxe présentant une liste d'actions de recette, consultezDéfinition de la structure d'une recette.

Example Exemple 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'
```

Pour plus d'informations sur l'utilisation de cette action de recette dans une opération d'API, consultez CreateRecipeou UpdateRecipe. Vous pouvez utiliser ces opérations d'API ainsi que d'autres dans votre propre code.

Étapes de la recette de structure des colonnes

Utilisez ces étapes de recette de structure de colonne pour modifier la structure de colonne de vos données.

Rubriques

OPÉRATION BOOLÉENNE

Créez une nouvelle colonne en fonction du résultat de la condition logique IF. Renvoie une valeur vraie si l'expression booléenne est vraie, une valeur fausse si l'expression booléenne est fausse, ou renvoie une valeur personnalisée.

Parameters

- `trueValueExpression`— Résultat lorsque la condition est remplie.
- `falseValueExpression`— Résultat lorsque la condition n'est pas remplie.
- `valueExpression`— Condition booléenne.
- `withExpressions`— Configuration pour les résultats agrégés.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Vous pouvez utiliser des valeurs constantes, des références de colonne et des résultats agrégés dans trueValueExpression, false ValueExpression et ValueExpression.

Example Exemple : valeurs constantes

Des valeurs qui restent inchangées, comme un chiffre ou une phrase.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Exemple : références de colonnes

Valeurs qui sont des colonnes dans le jeu de données.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Exemple : résultats agrégés

Valeurs calculées par des fonctions d'agrégation. Une fonction d'agrégation effectue un calcul sur une colonne et renvoie une valeur unique.

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

Les utilisateurs doivent convertir le JSON en chaîne en s'échappant.

Notez que les noms des paramètres dans true ValueExpressionValueExpression, false et ValueExpression doivent correspondre aux noms dans WithExpressions. Pour utiliser les résultats agrégés de certaines colonnes, vous devez créer des paramètres pour celles-ci et fournir les fonctions d'agrégation.

Example Exemple :

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
```

```
      }
    }
  }
}
```

Example Exemple : and/or

Vous pouvez utiliser et ou pour combiner plusieurs conditions.

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

Fonctions d'agrégation valides

Le tableau ci-dessous présente toutes les fonctions d'agrégation valides qui peuvent être utilisées dans une opération booléenne.

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
Numérique	Somme	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Renvoie la somme de `column.1`
	Mean	`:mean.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" } ]`	Renvoie la moyenne de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Écart absolu moyen	`:meanabsolutedeviation.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Renvoie l'écart absolu moyen de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Médiane	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Renvoie la médiane de `column.1`
	Produit (langue française non garantie)	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Renvoie le produit de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Écart-type standard	`:écart-standard. colonne.1`	`[ { "name": "standard deviation .column.1 ", "value": "stddev(` column.1` )", "type": "aggregat e" } ]`	Renvoie l'écart type de `column.1`
	Variance	`:varianc e.column.1`	`[ { "name": "variance .column.1 ", "value": "variance (`column. 1`)", "type": "aggregat e" } ]`	Renvoie la variance de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Erreur type de moyenne	`:erreur standard de mean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Renvoie l'erreur standard de la moyenne de `column.1`
	Asychité	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Renvoie l'asymétrie de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Renvoie le kurtosis de `column.1`
Datetime/ Numeric/Text	Nombre	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Renvoie le nombre total de lignes dans `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Compte distinct	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Renvoie le nombre total de lignes distinctes dans `column.1`
	Min	`:min.colonne.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Renvoie la valeur minimale de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Max	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Renvoie la valeur maximale de `column.1`

Conditions valides dans une ValueExpression

Le tableau ci-dessous indique les conditions prises en charge et les expressions de valeur que vous pouvez utiliser.

Type de colonne	Condition	Expression de valeur	Description
String	Contains	contient (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne contient du texte
	Ne contient pas	! contient (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne contient pas de texte

Type de colonne	Condition	Expression de valeur	Description
	Correspondance	correspondances (« colonne », « modèle »)	Condition pour tester si la valeur de la colonne correspond au modèle
	Ne correspond pas	! correspondances (« colonne », « modèle »)	Condition pour tester si la valeur de la colonne ne correspon d pas au modèle
	Starts with	startWith (`colonne`, « texte »)	Condition pour vérifier si la valeur de la colonne commence par du texte
	Ne commence pas par	! startWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne commence pas par du texte
	Termine par	EndsWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne se termine par du texte
	Ne se termine pas par	! EndsWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne se termine pas par du texte
Numérique	Inférieur à	`colonne` < numéro	Condition pour tester si la valeur de la colonne est inférieure à un nombre

Type de colonne	Condition	Expression de valeur	Description
	Inférieur ou égal à	`column` <= numéro	Condition pour tester si la valeur de la colonne est inférieure ou égale au nombre
	Supérieur à	`column` > numéro	Condition pour tester si la valeur de la colonne est supérieur e au nombre
	Supérieur ou égal à	`column` >= numéro	Condition pour tester si la valeur de la colonne est supérieur e ou égale au nombre
	Est comprise entre	isBetween (`colonne ', MinNumber, MaxNumber)	Condition pour tester si la valeur de la colonne est comprise entre minNumber et maxNumber
	N'est pas entre	! isBetween (`colonne ', MinNumber, MaxNumber)	Condition pour tester si la valeur de la colonne n'est pas comprise entre minNumber et maxNumber
Booléen	C'est vrai	`column` = VRAI	Condition pour tester si la valeur de la colonne est booléenne TRUE

Type de colonne	Condition	Expression de valeur	Description
	Est faux	`column` = FAUX	Condition pour tester si la valeur de la colonne est booléenne FALSE
Date/Timestamp	Plus tôt que	`colonne` < « date »	Condition pour tester si la valeur de la colonne est antérieure à la date
	Antérieur ou égal à	`colonne` <= « date »	Condition pour tester si la valeur de la colonne est antérieure ou égale à la date
	Plus tard que	`colonne` > « date »	Condition pour tester si la valeur de la colonne est postérieu re à la date
	Plus tard ou égal à	`colonne` >= « date »	Condition pour tester si la valeur de la colonne est ultérieure ou égale à la date
String/Numeric/Date/ Timestamp	C'est exactement	`column` = 'valeur'	Condition pour tester si la valeur de la colonne est exactement la valeur
	Is not (N’est pas)	`colonne` ! = « valeur »	Condition pour tester si la valeur de la colonne n'est pas une valeur

Type de colonne	Condition	Expression de valeur	Description
	Manque	IsMissing (`colonne')	Condition pour tester si la valeur de la colonne est manquante
	Ne manque pas	! IsMissing (`colonne')	Condition pour tester si la valeur de la colonne n'est pas manquante
	Est valide	isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne est valide (la valeur est de type donnée ou peut être convertie en type de données)
	N'est pas valide	! isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne n'est pas valide (la valeur est de type donnée ou peut être convertie en type de données)
Imbriqué	Manque	IsMissing (`colonne')	Condition pour tester si la valeur de la colonne est manquante
	Ne manque pas	! IsMissing (`colonne')	Condition pour tester si la valeur de la colonne n'est pas manquante

Type de colonne	Condition	Expression de valeur	Description
	Est valide	isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne est valide (la valeur est de type donnée ou peut être convertie en type de données)
	N'est pas valide	! isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne n'est pas valide (la valeur est de type donnée ou peut être convertie en type de données)

CAS_OPÉRATION

Créez une nouvelle colonne, basée sur le résultat de la condition logique CASE. L'opération de dossier passe en revue les conditions du cas et renvoie une valeur lorsque la première condition est remplie. Une fois qu'une condition est vraie, l'opération arrête la lecture et renvoie le résultat. Si aucune condition n'est vraie, elle renvoie la valeur par défaut.

Parameters

- `valueExpression`— Des conditions.
- `withExpressions`— Configuration pour les résultats agrégés.
- `targetColumn`— Nom de la colonne nouvellement créée.

Example Exemple

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

Fonctions d'agrégation valides

Le tableau ci-dessous présente toutes les fonctions d'agrégation valides qui peuvent être utilisées dans le cadre d'une opération de cas.

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
Numérique	Somme	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Renvoie la somme de `column.1`
	Mean	`:mean.column.1`	`[ { "name": "mean.column.1",`	Renvoie la moyenne de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée

			<pre>"value": "avg(`column.1`)",
			
			"type": "aggregate"
			}
			]</pre>	

	Écart absolu moyen	`:meanabsolutedeviation.column.1`	<pre>[
			{
			
			"name": "meanabsolutedeviation.column.1",
			
			"value": "mean_absolute_deviation(`column.1`)",
			
			"type": "aggregate"
			}
			]</pre>
				Renvoie l'écart absolu moyen de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Médiane	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Renvoie la médiane de `column.1`
	Produit (langue française non garantie)	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Renvoie le produit de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Écart-type standard	`:écart-standard. colonne.1`	`[ { "name": "standard deviation .column.1 ", "value": "stddev(` column.1` )", "type": "aggregat e" } ]`	Renvoie l'écart type de `column.1`
	Variance	`:varianc e.column.1`	`[ { "name": "variance .column.1 ", "value": "variance (`column. 1`)", "type": "aggregat e" } ]`	Renvoie la variance de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Erreur type de moyenne	`:erreur standard de mean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Renvoie l'erreur standard de la moyenne de `column.1`
	Asychité	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Renvoie l'asymétrie de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Renvoie le kurtosis de `column.1`
Datetime/ Numeric/Text	Nombre	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Renvoie le nombre total de lignes dans `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Compte distinct	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Renvoie le nombre total de lignes distinctes dans `column.1`
	Min	`:min.colonne.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Renvoie la valeur minimale de `column.1`

Type de colonne	Condition	Expression de valeur	Avec des expressions	Valeur renvoyée
	Max	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Renvoie la valeur maximale de `column.1`

Conditions valides dans une ValueExpression

Le tableau ci-dessous indique les conditions prises en charge et les expressions de valeur que vous pouvez utiliser.

Type de colonne	Condition	Expression de valeur	Description
String	Contains	contient (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne contient du texte
	Ne contient pas	! contient (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne contient pas de texte

Type de colonne	Condition	Expression de valeur	Description
	Correspondance	correspondances (« colonne », « modèle »)	Condition pour tester si la valeur de la colonne correspond au modèle
	Ne correspond pas	! correspondances (« colonne », « modèle »)	Condition pour tester si la valeur de la colonne ne correspon d pas au modèle
	Starts with	startWith (`colonne`, « texte »)	Condition pour vérifier si la valeur de la colonne commence par du texte
	Ne commence pas par	! startWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne commence pas par du texte
	Termine par	EndsWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne se termine par du texte
	Ne se termine pas par	! EndsWith (`colonne`, « texte »)	Condition pour tester si la valeur de la colonne ne se termine pas par du texte
Numérique	Inférieur à	`colonne` < numéro	Condition pour tester si la valeur de la colonne est inférieure à un nombre

Type de colonne	Condition	Expression de valeur	Description
	Inférieur ou égal à	`column` <= numéro	Condition pour tester si la valeur de la colonne est inférieure ou égale au nombre
	Supérieur à	`column` > numéro	Condition pour tester si la valeur de la colonne est supérieur e au nombre
	Supérieur ou égal à	`column` >= numéro	Condition pour tester si la valeur de la colonne est supérieur e ou égale au nombre
	Est comprise entre	isBetween (`colonne ', MinNumber, MaxNumber)	Condition pour tester si la valeur de la colonne est comprise entre minNumber et maxNumber
	N'est pas entre	! isBetween (`colonne ', MinNumber, MaxNumber)	Condition pour tester si la valeur de la colonne n'est pas comprise entre minNumber et maxNumber
Booléen	C'est vrai	`column` = VRAI	Condition pour tester si la valeur de la colonne est booléenne TRUE

Type de colonne	Condition	Expression de valeur	Description
	Est faux	`column` = FAUX	Condition pour tester si la valeur de la colonne est booléenne FALSE
Date/Timestamp	Plus tôt que	`colonne` < « date »	Condition pour tester si la valeur de la colonne est antérieure à la date
	Antérieur ou égal à	`colonne` <= « date »	Condition pour tester si la valeur de la colonne est antérieure ou égale à la date
	Plus tard que	`colonne` > « date »	Condition pour tester si la valeur de la colonne est postérieu re à la date
	Plus tard ou égal à	`colonne` >= « date »	Condition pour tester si la valeur de la colonne est ultérieure ou égale à la date
String/Numeric/Date/ Timestamp	C'est exactement	`column` = 'valeur'	Condition pour tester si la valeur de la colonne est exactement la valeur
	Is not (N’est pas)	`colonne` ! = « valeur »	Condition pour tester si la valeur de la colonne n'est pas une valeur

Type de colonne	Condition	Expression de valeur	Description
	Manque	IsMissing (`colonne')	Condition pour tester si la valeur de la colonne est manquante
	Ne manque pas	! IsMissing (`colonne')	Condition pour tester si la valeur de la colonne n'est pas manquante
	Est valide	isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne est valide (la valeur est de type donnée ou peut être convertie en type de données)
	N'est pas valide	! isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne n'est pas valide (la valeur est de type donnée ou peut être convertie en type de données)
Imbriqué	Manque	IsMissing (`colonne')	Condition pour tester si la valeur de la colonne est manquante
	Ne manque pas	! IsMissing (`colonne')	Condition pour tester si la valeur de la colonne n'est pas manquante

Type de colonne	Condition	Expression de valeur	Description
	Est valide	isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne est valide (la valeur est de type donnée ou peut être convertie en type de données)
	N'est pas valide	! isValid (`column`, type de données)	Condition pour tester si la valeur de la colonne n'est pas valide (la valeur est de type donnée ou peut être convertie en type de données)

FLAG_COLUMN_FROM_NULL

Crée une nouvelle colonne en fonction de la présence de valeurs nulles dans une colonne existante.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn`— Nom de la nouvelle colonne à créer.
- `flagType`— Une valeur qui doit être définie sur`Null values`.
- `trueString`— Une valeur pour la nouvelle colonne, si une valeur nulle est trouvée dans la source. Si aucune valeur n'est spécifiée, la valeur par défaut est `True`.
- `falseString`— Une valeur pour la nouvelle colonne, si une valeur non nulle est trouvée dans la source. Si aucune valeur n'est spécifiée, la valeur par défaut est `False`.

Example Exemple

```
{
```

```
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

Crée une nouvelle colonne en fonction de la présence d'un modèle défini par l'utilisateur dans une colonne existante.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn`— Nom de la nouvelle colonne à créer.
- `flagType`— Une valeur qui doit être définie sur`Pattern`.
- `pattern`— Expression régulière indiquant le modèle à évaluer.
- `trueString`— Une valeur pour la nouvelle colonne, si une valeur nulle est trouvée dans la source. Si aucune valeur n'est spécifiée, la valeur par défaut est `True`.
- `falseString`— Une valeur pour la nouvelle colonne, si une valeur non nulle est trouvée dans la source. Si aucune valeur n'est spécifiée, la valeur par défaut est `False`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
```

```
    }
}
```

MERGE

Fusionne deux colonnes ou plus dans une nouvelle colonne.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste d'une ou de plusieurs colonnes à fusionner.
- `delimiter`— Séparateur optionnel entre les valeurs, à afficher dans la colonne cible.
- `targetColumn`— Nom de la colonne fusionnée à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_DELIMITER

Divise une colonne en trois nouvelles colonnes, selon un séparateur de début et de fin.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `patternOption1`— JSON-encoded Chaîne représentant un ou plusieurs caractères indiquant le premier délimiteur.
- `patternOption2`— JSON-encoded Chaîne représentant un ou plusieurs caractères indiquant le second délimiteur.

- `pattern`— Un ou plusieurs caractères à utiliser comme séparateur lors du fractionnement des données.
- `includeInSplit`— Si vrai, inclut le modèle dans la nouvelle colonne ; dans le cas contraire, le modèle est supprimé.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_DELIMITER",
        "Parameters": {
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_POSITIONS

Divise une colonne en trois nouvelles colonnes, selon les décalages que vous spécifiez.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `startPosition`— Position du personnage à laquelle la division doit commencer.
- `endPosition`— Position du personnage à laquelle la division doit se terminer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
```

```
        }
    }
}
```

SPLIT_COLUMN_FROM_END

Divise une colonne en deux nouvelles colonnes, décalées par rapport à la fin de la chaîne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `position`— Position du caractère, à partir de l'extrémité droite de la chaîne, où le clivage doit se produire.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
}
```

SPLIT_COLUMN_FROM_START

Divise une colonne en deux nouvelles colonnes, décalées par rapport au début de la chaîne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `position`— Position du caractère, à partir de l'extrémité gauche de la chaîne, où le clivage doit se produire.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

SPLIT_COLUMN_MULTIPLE_DELIMITER

Divise une colonne en fonction de plusieurs délimiteurs.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `patternOptions`— JSON-encoded Chaîne représentant un ou plusieurs modèles qui déterminent les critères de division.
- `pattern`— Un ou plusieurs caractères à utiliser comme séparateur lors du fractionnement des données.
- `limit`— Combien de splits effectuer. Le minimum est de 1 ; le maximum est de 20.
- `includeInSplit`— Si vrai, inclut le modèle dans la nouvelle colonne ; dans le cas contraire, le modèle est supprimé.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
}
```

SPLIT_COLUMN_SINGLE_DELIMITER

Divise une colonne en une ou plusieurs nouvelles colonnes, selon un délimiteur spécifique.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Un ou plusieurs caractères à utiliser comme séparateur lors du fractionnement des données.
- `limit`— Combien de splits effectuer. Le minimum est de 1 ; le maximum est de 20.
- `includeInSplit`— Si vrai, inclut le modèle dans la nouvelle colonne ; dans le cas contraire, le modèle est supprimé.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_SINGLE_DELIMITER",
        "Parameters": {
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVAL

Divise une colonne à des intervalles de n caractères, où vous spécifiez n.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `startPosition`— Position du personnage à laquelle la division doit commencer.
- `interval`— Le nombre de caractères à ignorer avant le prochain split.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

Étapes de la recette de mise en forme

Utilisez les étapes de la recette de formatage des colonnes pour modifier le format des données de vos colonnes.

Rubriques

- NOMBRE_FORMAT
- FORMAT_NUMÉRO_TÉLÉPHONE

NOMBRE_FORMAT

Renvoie une colonne dans laquelle une valeur numérique est convertie en chaîne formatée.

Parameters

- `sourceColumn` – String. Le nom d'une colonne existante.
- `decimalPlaces`— Entier. La valeur du nombre de chiffres après le séparateur décimal.
- `numericDecimalSeparator` – String. L'une des valeurs suivantes indiquant le séparateur décimal :
  - "."
  - ","
- `numericThousandSeparator` – String. L'une des valeurs suivantes indiquant le séparateur de milliers :
  - nul. Indique que le séparateur de milliers n'est pas activé.

  - ","
  - " "
  - "."
  - "\\"
- `numericAbbreviatedUnit` – String. L'une des valeurs suivantes indiquant l'unité d'abréviation :
  - nul. Indique qu'aucune unité d'abréviation n'est activée.
  - « MILLE »
  - « MILLION »
  - « MILLIARD »
  - « BILLION »
- `numericUnitAbbreviation` – String. L'une des valeurs suivantes ou toute valeur personnalisée indiquant l'abréviation de l'unité :
  - nul. Indique que l'abréviation des unités n'est pas activée.
  -

Unité d'abréviation	Options
Milliers	K, k, M, mille, personnalisé
Millions	M, m, MM, million, personnalisé
Milliard	B, milliard, milliard, personnalisé
Trillions	T, dix, billions, personnalisé

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
```

```
            "numericUnitAbbreviation": "K"
        }
    }
}
```

FORMAT_NUMÉRO_TÉLÉPHONE

Renvoie une colonne dans laquelle une chaîne de numéro de téléphone est convertie en valeur formatée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `phoneNumberFormat` : format dans lequel le numéro de téléphone sera converti. Si aucun format n'est spécifié, le format par défaut est `E.164`, un format de numéro de téléphone standard reconnu à l'échelle internationale. Les valeurs valides sont les suivantes :
  - `E164`(omettez la période suivante) `E`
- `defaultRegion` : code de région valide composé de deux ou trois lettres majuscules qui indique la région du numéro de téléphone lorsque aucun code de pays n'est présent dans le numéro lui-même. Tout au plus, une des `defaultRegion` ou `defaultRegionColumn` peut être fournie.
- `defaultRegionColumn`— Le nom d'une colonne de [type de données avancé](#)`Country`. Le code de région de la colonne spécifiée est utilisé pour déterminer le code de pays pour le numéro de téléphone lorsque aucun code de pays n'est présent dans le numéro lui-même. Tout au plus, une des `defaultRegion` ou `defaultRegionColumn` peut être fournie.

Remarques

- Les entrées qui ne peuvent pas être formatées selon un numéro de téléphone valide restent inchangées.
- Si aucune région par défaut n'est fournie et qu'un numéro de téléphone ne commence pas par le symbole plus (+) et le code du pays d'appel, le numéro de téléphone n'est pas formaté.

Example

Exemple : région par défaut fixe

```
{
```

```
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

Exemple : option de colonne de région par défaut

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

Étapes de recette de structure de données

Utilisez ces étapes de recette pour tabuler et résumer les données sous différents angles, ou pour exécuter des fonctions avancées.

Rubriques

- NID DANS UN TABLEAU
- NEST_TO_MAP
- DU NID À LA STRUCTURE
- UNNEST_ARRAY
- UNNEST_MAP
- UNNEST_STRUCT
- UNNEST_STRUCT_N
- GROUP_BY
- JOIN

- PIVOT
- SCALE
- TRANSPOSER
- UNION
- UNPIVOT

NID DANS UN TABLEAU

Convertit les colonnes sélectionnées par l'utilisateur en valeurs de tableau. L'ordre des colonnes sélectionnées est conservé lors de la création du tableau résultant. Les différents types de données de colonne sont convertis en un type commun qui prend en charge les types de données de toutes les colonnes.

Parameters

- `sourceColumns`— Liste des colonnes sources.
- `targetColumn`— Nom de la colonne cible.
- `removeSourceColumns`— Contient la valeur `true` ou `false` indique si l'utilisateur souhaite supprimer les colonnes source sélectionnées.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_MAP

Convertit les colonnes sélectionnées par l'utilisateur en paires clé-valeur, chacune comportant une clé représentant le nom de la colonne et une valeur représentant la valeur de la ligne. L'ordre de la

colonne sélectionnée n'est pas conservé lors de la création de la carte résultante. Les différents types de données de colonne sont convertis en un type commun qui prend en charge les types de données de toutes les colonnes.

Parameters

- `sourceColumns`— Liste des colonnes sources.
- `targetColumn`— Nom de la colonne cible.
- `removeSourceColumns`— Contient la valeur `true` ou `false` indique si l'utilisateur souhaite supprimer les colonnes source sélectionnées.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

DU NID À LA STRUCTURE

Convertit les colonnes sélectionnées par l'utilisateur en paires clé-valeur, chacune comportant une clé représentant le nom de la colonne et une valeur représentant la valeur de la ligne. L'ordre des colonnes sélectionnées et le type de données de chaque colonne sont conservés dans la structure résultante.

Parameters

- `sourceColumns`— Liste des colonnes sources.
- `targetColumn`— Nom de la colonne cible.
- `removeSourceColumns`— Contient la valeur `true` ou `false` indique si l'utilisateur souhaite supprimer les colonnes source sélectionnées.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

UNNEST_ARRAY

Désimbrique une colonne de type `array` dans une nouvelle colonne. Si le tableau contient plusieurs valeurs, une ligne correspondant à chaque élément est générée. Cette fonction désimbrique uniquement un niveau d'une colonne de tableau.

Parameters

- `sourceColumn`— Le nom d'une colonne existante. Cette colonne doit être de `struct` type.
- `targetColumn`— Nom de la colonne cible générée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_MAP

Désimbrique une colonne de type `map` et génère une colonne pour la clé et la valeur. S'il existe plusieurs paires clé-valeur, une ligne correspondant à chaque valeur clé sera générée. Cette fonction ne désimbrique qu'un seul niveau d'une colonne de carte.

Parameters

- `sourceColumn`— Le nom d'une colonne existante. Cette colonne doit être de `struct` type.
- `removeSourceColumn`— Si`true`, la colonne source est supprimée une fois la fonction terminée.
- `targetColumn`— Si elle est fournie, chacune des colonnes générées commencera par ce préfixe.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

Désimbriquez une colonne de type `struct` et générez une colonne pour chacune des clés présentes dans la structure. Cette fonction désactive uniquement le niveau 1 de la structure.

Parameters

- `sourceColumn`— Le nom d'une colonne existante. Cette colonne doit être de type structure.
- `removeSourceColumn`— Si`true`, la colonne source est supprimée une fois la fonction terminée.
- `targetColumn`— Si elle est fournie, chacune des colonnes générées commencera par ce préfixe.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

Crée une nouvelle colonne pour chaque champ d'une colonne de type sélectionnée`struct`.

Par exemple, étant donné la structure suivante :

```
          user {
             name: “Ammy”
             address: {
                state: "CA",
                zipcode: 12345
             }
          }
```

Cette fonction crée 3 colonnes :

nom d'utilisateur	utilisateur.adresse.state	utilisateur.adresse.code postal
Ammy	CA	12345

Parameters

- `sourceColumns`— Liste des colonnes sources.
- `regexColumnSelector`— Expression régulière permettant de sélectionner les colonnes à désimbriquer.

- `removeSourceColumn`— Valeur booléenne. Si vrai, supprimez la colonne source ; sinon, conservez-la.
- `unnestLevel`— Le nombre de niveaux à dénicher.
- `delimiter`— Le délimiteur est utilisé dans le nom de colonne nouvellement créé pour séparer les différents niveaux de la structure. Par exemple : si le délimiteur est «/», le nom de la colonne sera sous la forme suivante : « user/address /state ».
- `conditionExpressions`— Expressions conditionnelles.

Example Exemple

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

GROUP_BY

Résume les données en groupant les lignes par une ou plusieurs colonnes, puis en appliquant une fonction d'agrégation à chaque groupe.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes constituant la base de chaque groupe.
- `groupByAggFunctions`— JSON-encoded Chaîne représentant la liste des fonctions d'agrégation à appliquer. (Si vous ne souhaitez pas d'agrégation, spécifiez`UNAGGREGATED`.)
- `useNewDataFrame`— Si c'est vrai, les résultats de GROUP_BY sont disponibles dans la session du projet, en remplacement de leur contenu actuel.

Example Exemple

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
      }
    }
  }
]
```

JOIN

Effectue une opération de jointure sur deux ensembles de données.

Parameters

- `joinKeys`— JSON-encoded Chaîne représentant une liste de colonnes de chaque ensemble de données devant servir de clés de jointure.
- `joinType`— Type de jointure à effectuer. Doit être l'un des suivants : `INNER_JOIN` `LEFT_JOIN` `RIGHT_JOIN` | | `OUTER_JOIN` | `LEFT_EXCLUDING_JOIN` | `RIGHT_EXCLUDING_JOIN` | `OUTER_EXCLUDING_JOIN`
- `leftColumns`— JSON-encoded Chaîne représentant une liste de colonnes de l'ensemble de données actif actuel.
- `rightColumns`— JSON-encoded Chaîne représentant une liste de colonnes d'un autre ensemble de données (secondaire) à joindre à l'ensemble de données actuel.
- `secondInputLocation`— Une URL Amazon S3 qui renvoie au fichier de données de l'ensemble de données secondaire.
- `secondaryDatasetName`— Le nom du jeu de données secondaire.

Example Exemple

```
{
```

```
    "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

Convertit toutes les valeurs de ligne d'une colonne sélectionnée en colonnes individuelles contenant des valeurs.

Parameters

- `sourceColumn`— Le nom d'une colonne existante. La colonne peut avoir un maximum de 10 valeurs distinctes.
- `valueColumn`— Le nom d'une colonne existante. La colonne peut avoir un maximum de 10 valeurs distinctes.
- `aggregateFunction`— Nom d'une fonction d'agrégation. Si vous ne souhaitez pas d'agrégation, utilisez le mot clé`COLLECT_LIST`.

Example Exemple

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
            "valueColumn": "all_votes"
        }
    }
}
```

SCALE

Redimensionne ou normalise la plage de données d'une colonne numérique.

Parameters

- `sourceColumn`— Le nom d'une colonne existante.
- `strategy`— L'opération à appliquer aux valeurs des colonnes :
  - `MIN_MAX`— Redimensionne les valeurs dans une plage de [0,1].
  - `SCALE_BETWEEN`— Redimensionne les valeurs dans une plage de deux valeurs spécifiées.
  - `MEAN_NORMALIZATION`— Redimensionne les données pour avoir une moyenne (μ) de 0 et un écart type (σ) de 1 dans une plage de [-1, 1].
  - `Z_SCORE`— Échelle linéairement les valeurs des données pour qu'elles aient une moyenne (μ) de 0 et un écart type (σ) de 1. Idéal pour gérer les valeurs aberrantes.
- `targetColumn`— Le nom de la colonne qui doit contenir les résultats.

Example Exemple

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
```

```
    }
}
```

TRANSPOSER

Convertit toutes les lignes sélectionnées en colonnes et les colonnes en lignes.

Column 1	Column A	Column B	Column C
Row A	Value A	Value B	Value C
Row B	Value A1	Value B1	Value C1

New column	Row A	Row B
Column A	Value A	Value A1
Column B	Value B	Value B1
Column C	Value C	Value C1

Parameters

- `pivotColumns`— JSON-encoded Chaîne représentant une liste de colonnes dont les lignes seront converties en noms de colonnes.
- `valueColumns`— JSON-encoded Chaîne représentant une liste d'une ou plusieurs colonnes à convertir en lignes.
- `aggregateFunction`— Nom d'une fonction d'agrégation. Si vous ne souhaitez pas d'agrégation, utilisez le mot clé`COLLECT_LIST`.
- `newColumn`— La colonne qui contient les colonnes transposées sous forme de valeurs.

Example Exemple

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
```

```
            "newColumn": "Student"
        }
    }

}
```

UNION

Combine les lignes de deux ensembles de données ou plus en un seul résultat.

Parameters

- `datasetsColumns`— JSON-encoded Chaîne représentant la liste de toutes les colonnes des ensembles de données.
- `secondaryDatasetNames`— JSON-encoded Chaîne représentant une liste d'un ou de plusieurs ensembles de données secondaires.
- `secondaryInputs`— Une JSON-encoded chaîne représentant une liste de buckets Amazon S3 et de noms de clés d'objets indiquant DataBrew où trouver le ou les ensembles de données secondaires.
- `targetColumnNames`— JSON-encoded Chaîne représentant une liste de noms de colonnes pour les résultats.

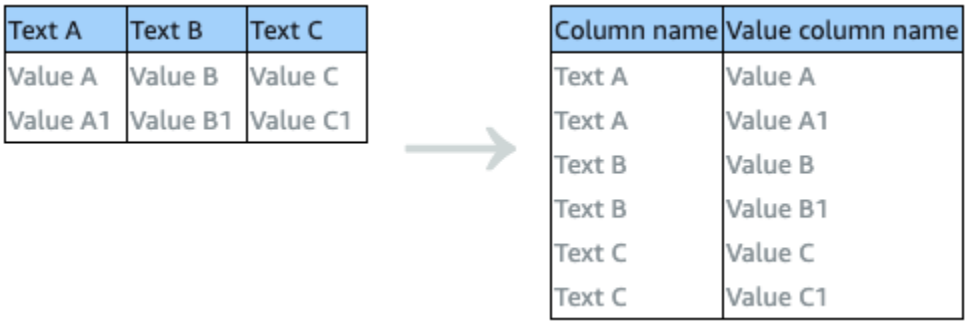
Example Exemple

```
{
    "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

Convertit toutes les valeurs de colonne d'une ligne sélectionnée en lignes individuelles contenant des valeurs.



Text A	Text B	Text C
Value A	Value B	Value C
Value A1	Value B1	Value C1

Column name	Value column name
Text A	Value A
Text A	Value A1
Text B	Value B
Text B	Value B1
Text C	Value C
Text C	Value C1

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste d'une ou de plusieurs colonnes à annuler.
- `unpivotColumn`— Colonne de valeurs pour l'opération de dépivotement.
- `valueColumn`— Colonne contenant les valeurs non pivotantes.

Example Exemple

```
{
    "Action": {
        "Operation": "UNPIVOT",
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
    }
}
```

Étapes de la recette de science des données

Utilisez ces étapes de recette pour tabuler et résumer les données sous différents angles, ou pour effectuer des transformations avancées.

Rubriques

BINARISATION

Prend toutes les valeurs d'une colonne source numérique sélectionnée, les compare à une valeur de seuil et génère une nouvelle colonne avec un 1 ou un 0 pour chaque ligne.

Parameters

- `sourceColumn` : nom d’une colonne existante.

  `targetColumn` : le nom de la nouvelle colonne à créer.

  `threshold`— Numéro indiquant le seuil d'attribution de la valeur 0 ou 1.

  `flip`— Option permettant d'inverser l'assignation binaire afin que les valeurs inférieures soient attribuées à 1 et les valeurs supérieures à 0. Lorsque le paramètre flip est vrai, les valeurs inférieures ou égales à la valeur de seuil donnent 1, et les valeurs supérieures à la valeur de seuil donnent 0.

Example Exemple

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

```
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
}
```

BUKETISATION

La mise en compartiments (appelée Binning dans la console) prend les éléments d'une colonne de valeurs numériques, les regroupe dans des groupes définis par des plages numériques et génère une nouvelle colonne qui affiche le groupe pour chaque ligne. La segmentation peut être effectuée à l'aide de divisions ou de pourcentages. Le premier exemple ci-dessous utilise des divisions et le second un pourcentage.

Parameters

- `sourceColumn` : nom d'une colonne existante.

  `targetColumn` : le nom de la nouvelle colonne à créer.

  `bucketNames`— Liste des noms de compartiments.

  `splits`— Liste des niveaux de compartiment. Les buckets sont consécutifs, et la limite supérieure d'un bucket sera la borne inférieure du bucket suivant.

  `percentage`— Chaque compartiment sera décrit sous forme de pourcentage.

Example Exemple d'utilisation de splits

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
```

```
        }
    }
}
```

Example Exemple utilisant un pourcentage

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

MAPPAGE_CATÉGORIQUE

Associe une ou plusieurs valeurs catégoriques à des valeurs numériques ou autres

Parameters

- `sourceColumn` : nom d'une colonne existante.

  `categoryMap`— JSON-encoded Chaîne représentant une carte entre les valeurs et les catégories.

  `deleteOtherRows`— Dans `true` ce cas, toutes les lignes non mappées seront supprimées de l'ensemble de données.

  `other`— Lorsqu'elles sont fournies, toutes les valeurs non mappées seront remplacées par cette valeur.

  `keepOthers`— Si c'est vrai, toutes les valeurs non mappées resteront les mêmes.

  `mapType`— Type de données de la colonne mappée.

  `targetColumn`— Le nom de la colonne qui doit contenir les résultats.

Example Exemple

```
{
    "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

Crée n colonnes numériques, où n est le nombre de valeurs uniques dans une variable catégorielle sélectionnée.

Prenons l'exemple d'une colonne nomméeshirt_size. Les chemises sont disponibles en taille petite, moyenne, grande ou très grande. Les données de colonne peuvent ressembler à ce qui suit.

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
M
```

Dans ce scénario, il existe quatre valeurs distinctes pourshirt_size. ONE_HOT_ENCODINGGénère donc quatre nouvelles colonnes. Chaque nouvelle colonne est nomméeshirt_size_*x*, où *x* représente une shirt_size valeur distincte.

Les résultats de `shirt_size` et les quatre colonnes générées ressemblent à ceci.

```
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 colonne que vous spécifiez `ONE_HOT_ENCODING` peut comporter au maximum dix (10) valeurs distinctes.

Parameters

- `sourceColumn` : nom d'une colonne existante. La colonne peut avoir un maximum de 10 valeurs distinctes.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SCALE

Redimensionne ou normalise la plage de données d'une colonne numérique.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `strategy`— L'opération à appliquer aux valeurs des colonnes :
  - `MIN_MAX`— Redimensionne les valeurs dans une plage de [0,1]
  - `SCALE_BETWEEN`— Redimensionne les valeurs dans une plage de 2 valeurs spécifiées.
  - `MEAN_NORMALIZATION`— Redimensionne les données pour avoir une moyenne (μ) de 0 et un écart type (σ) de 1 dans une plage de [-1, 1]
  - `Z_SCORE`— Ajustez linéairement les valeurs des données pour qu'elles aient une moyenne (μ) de 0 et un écart type (σ) de 1. Idéal pour gérer les valeurs aberrantes.
- `targetColumn`— Le nom de la colonne qui doit contenir les résultats.

Example Exemple

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

ASYMÉTRIE

Applique des transformations aux valeurs de vos données pour modifier la forme de distribution et son inclinaison.

Parameters

- `sourceColumn` : nom d'une colonne existante.

  `targetColumn` : le nom de la nouvelle colonne à créer.

  `skewFunction`
  - `ROOT`— extrait la racine de valeur. La racine peut être fournie dans le `value` paramètre.

`LOG`— valeur de base du journal. La base du log peut être fournie dans le `value` paramètre.

`SQUARE`— fonction carrée

`value`— Argument de la fonction SkewFunction.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

TOKENISATION

Divise le texte en unités plus petites, ou jetons, tels que des mots ou des termes individuels.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `delimiter`— Un séparateur personnalisé qui apparaît entre les mots tokenisés. (Le comportement par défaut consiste à séparer chaque jeton par un espace.)
- `expandContractions`— Si`ENABLED`, développe les mots contractés. Par exemple : « ne pas » devient « ne pas ».
- `stemmingMode`— Divise le texte en unités ou en jetons plus petits, tels que des mots ou termes minuscules individuels. Deux modes de découpage sont disponibles : `PORTER` |`LANCASTER`.
- `stopWordRemovalMode`— Supprime les mots courants tels que a, an, le, etc.
- `customStopWords`— Pour`StopWordRemovalMode`, vous permet de définir une liste personnalisée de mots vides.

- `targetColumn`— Le nom de la colonne qui doit contenir les résultats.

Example Exemple

```
{
    "Action": {
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

Fonctions mathématiques

Vous trouverez ci-dessous des rubriques de référence pour les fonctions mathématiques qui fonctionnent avec des actions de recette.

Rubriques

- ABSOLUTE
- ADD
- CEILING
- DEGREES
- DIVISER
- EXPOSANT
- FLOOR
- EST_PAIR
- EST ÉTRANGE

- LN
- LOG
- MOD
- MULTIPLIER
- NIER
- PI
- POWER
- RADIANS
- ALEATOIRE
- RANDOM_BETWEEN
- ROUND
- SIGN
- RACINE CARRÉE
- SOUSTRAIRE

ABSOLUTE

Renvoie la valeur absolue du nombre saisi dans une nouvelle colonne. La valeur absolue est la distance entre le nombre et zéro, qu'il soit positif ou négatif

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
```

```
        }
    }
}
```

ADD

Additionne les valeurs des colonnes d'entrée dans une nouvelle colonne, en utilisant (`sourceColumn1+sourceColumn2`) ou (`sourceColumn1+value1`).

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `value1`— Une valeur numérique.
- `sourceColumn2` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

CEILING

Renvoie le plus petit nombre entier supérieur ou égal aux nombres décimaux d'entrée dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value1`— Une valeur numérique.

- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "CEILING",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

DEGREES

Convertit les radians d'un angle en degrés et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

DIVISER

Divise un nombre saisi par un autre et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `value1`— Une valeur numérique.
- `sourceColumn2` : nom d'une colonne existante.
- `value2`— Une valeur numérique.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
        }
    }
}
```

EXPOSANT

Renvoie le nombre d'Euler élevé au nième degré dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
```

```
            "targetColumn": "age_EXPONENT"
        }
    }
}
```

FLOOR

Renvoie le plus grand nombre entier supérieur ou égal au nombre saisi dans une nouvelle colonne.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `value`— Une valeur numérique.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

EST_PAIR

Renvoie une valeur booléenne dans une nouvelle colonne qui indique si la colonne ou la valeur source est paire. Si la colonne ou la valeur source est décimale, le résultat est false.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `trueString` : chaîne indiquant si la valeur est paire.
- `falseString`— Chaîne indiquant si la valeur n'est pas paire.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
        "Parameters": {
            "falseString": "Value is odd",
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

EST ÉTRANGE

Renvoie une valeur booléenne dans une nouvelle colonne qui indique si la colonne ou la valeur source est impaire. Si la colonne ou la valeur source est décimale, le résultat est false.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `trueString`— Chaîne indiquant si la valeur est impaire.
- `falseString`— Chaîne qui indique si la valeur n'est pas impaire.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
}
```

LN

Renvoie le logarithme naturel (nombre d'Euler) d'une valeur dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
        }
    }
}
```

LOG

Renvoie le logarithme d'une valeur dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `base`— La base du logarithme. La valeur par défaut est 10.

Example Exemple

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

```
            "base": "10",
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
    }
}
```

MOD

Renvoie le pourcentage qu'un nombre représente par rapport à un autre nombre dans une nouvelle colonne.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `sourceColumn2` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

MULTIPLIER

Multiplie deux nombres et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `value1`— Une valeur numérique.

- `sourceColumn2` : nom d'une colonne existante.
- `value2`— Une valeur numérique.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

NIER

Annule une valeur et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

Renvoie la valeur de pi (3,141592653589793) dans une nouvelle colonne.

Parameters

- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

POWER

Renvoie la valeur d'un nombre à la puissance de l'exposant dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `value`— Nombre dont la valeur doit être augmentée.
- `targetColumn` : le nom de la nouvelle colonne à créer.
- `exponent`— La puissance à laquelle la valeur sera augmentée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "POWER",
        "Parameters": {
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
    }
}
```

RADIANS

Convertit les degrés en radians (divise par 180/pi) et renvoie la valeur dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

ALEATOIRE

Renvoie un nombre aléatoire compris entre 0 et 1 dans une nouvelle colonne.

Parameters

- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "RANDOM",
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

RANDOM_BETWEEN

Dans une nouvelle colonne, renvoie un nombre aléatoire compris entre une limite inférieure spécifiée (inclus) et une limite supérieure spécifiée (inclus).

Parameters

- `lowerBound`— Limite inférieure de la plage de nombres aléatoires.
- `upperBound`— Limite supérieure de la plage de nombres aléatoires.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

ROUND

Arrondit une valeur numérique à l'entier le plus proche dans une nouvelle colonne. Elle est arrondie lorsque la fraction est égale ou supérieure à 0,5.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

SIGN

Renvoie une nouvelle colonne avec -1 si la valeur est inférieure à 0, 0 si la valeur est 0 et +1 si la valeur est supérieure à 0.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

RACINE CARRÉE

Renvoie la racine carrée d'une valeur dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

SOUSTRAIRE

Soustrait un nombre d'un autre et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `value1`— Une valeur numérique.
- `sourceColumn2` : nom d'une colonne existante.
- `value2`— Une valeur numérique.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
```

```
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "targetColumn": "weight_minus_10_kg",
            "value2": "10"
        }
    }
}
```

Fonctions d’agrégation

Vous trouverez ci-dessous des rubriques de référence pour les fonctions d'agrégation qui fonctionnent avec des actions de recette.

Rubriques

- ANY
- AVERAGE
- COUNT
- NOMBRE_DISTINCT
- KTH_LARGEST
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN
- MODE
- ÉCART-TYPE
- SUM
- ÉCART

ANY

Renvoie toutes les valeurs des colonnes source sélectionnées dans une nouvelle colonne. Les valeurs vides et nulles sont ignorées.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

AVERAGE

Calcule la moyenne des valeurs des colonnes source et renvoie le résultat dans une nouvelle colonne. Toute valeur non numérique est ignorée.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

COUNT

Renvoie le nombre de valeurs des colonnes source sélectionnées dans une nouvelle colonne. Les valeurs vides et nulles sont ignorées.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

NOMBRE_DISTINCT

Renvoie le nombre total de valeurs distinctes des colonnes source sélectionnées dans une nouvelle colonne. Les valeurs vides et nulles sont ignorées.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

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

```
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
    }
}
```

KTH_LARGEST

Renvoie le k e plus grand nombre des colonnes source sélectionnées dans une nouvelle colonne.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.
- `value`— Un nombre représentant k.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

Renvoie le k e plus grand nombre unique des colonnes source sélectionnées dans une nouvelle colonne.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

  `value`— Un nombre représentant k.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

Renvoie la valeur numérique maximale des colonnes source sélectionnées dans une nouvelle colonne. Toute valeur non numérique est ignorée.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
}
```

MEDIAN

Renvoie la médiane, le chiffre du milieu d'un groupe trié de nombres, à partir des colonnes source sélectionnées dans une nouvelle colonne. Toute valeur non numérique est ignorée.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

Renvoie la valeur minimale des colonnes source sélectionnées dans une nouvelle colonne. Toute valeur non numérique est ignorée.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MIN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

Renvoie le mode, le nombre qui apparaît le plus souvent, à partir des colonnes source sélectionnées dans une nouvelle colonne. Toute valeur non numérique est ignorée. Pour plusieurs modes, le mode est calculé à l'aide de la fonction modale.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

ÉCART-TYPE

Renvoie l'écart type par rapport aux colonnes source sélectionnées dans une nouvelle colonne.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

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

```
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

SUM

Renvoie la somme des valeurs des colonnes source sélectionnées dans une nouvelle colonne. Toute valeur non numérique est traitée comme 0.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
    }
}
```

ÉCART

Renvoie la variation par rapport aux colonnes source sélectionnées dans une nouvelle colonne. La variance est définie comme`Var(X) = [Sum ((X – mean(X))^2)]/Count(X)`.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant une liste de colonnes existantes.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "VARIANCE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

Fonctions de texte

Vous trouverez ci-dessous des rubriques de référence pour les fonctions de texte qui fonctionnent avec des actions de recette.

Rubriques

- CHAR
- ENDS_WITH
- EXACT
- TROUVER
- LEFT
- LEN
- LOWER
- FUSIONNER_COLONNES_ET_VALEURS
- CORRECT
- SUPPRIMER_SYMBOLES
- SUPPRIMER_WHITESPACE
- CHAÎNE DE RÉPÉTITION
- RIGHT
- RIGHT_FIND
- STARTS_WITH

- STRING SUPÉRIEUR À
- STRING_GREATER_THAN_EQUAL
- CHAÎNE INFÉRIEURE À
- CHAÎNE INFÉRIEURE À ÉGALE
- SUBSTRING
- TRIM
- UNICODE
- UPPER

CHAR

Renvoie dans une nouvelle colonne le caractère Unicode pour chaque entier de la colonne source ou pour une valeur entière personnalisée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value`— Un entier qui représente une valeur Unicode.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
```

```
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
}
```

ENDS_WITH

Renvoie `true` une nouvelle colonne si un nombre spécifié de caractères situés le plus à droite, ou une chaîne personnalisée, correspond à un modèle.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `pattern`— Expression régulière qui doit correspondre à la fin de la chaîne.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
```

```
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

EXACT

Crée une nouvelle colonne contenant l'un des éléments suivants :

- `True`si une chaîne d'une colonne (ou d'une valeur) correspond exactement à une autre chaîne d'une autre colonne (ou valeur).
- `False`s'il n'y a pas de correspondance.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `sourceColumn2` : nom d'une colonne existante.
- `value1` : chaîne de caractères à évaluer.
- `value2` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous ne pouvez spécifier qu'une seule des combinaisons suivantes :
>
> - Les deux`sourceColumn`*N*.
> - L'un des `sourceColumn`*N* et l'un des`value`*N*.
> - Les deux`value`*N*.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "EXACT",
```

```
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

TROUVER

En effectuant une recherche de gauche à droite, vous trouvez les chaînes correspondant à une chaîne spécifiée dans la colonne source ou à partir d'une valeur personnalisée, puis renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Expression régulière à rechercher.
- `position`— Position du caractère par laquelle commencer, à partir de l'extrémité gauche de la chaîne.
- `ignoreCase`— Si`true`, ignorez les différences de majuscules (entre majuscules et minuscules) entre les lettres. Pour appliquer une correspondance stricte, utilisez `false` plutôt.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

LEFT

À partir d'un certain nombre de caractères, prend le nombre de caractères le plus à gauche de la chaîne de la colonne source ou de la chaîne personnalisée, et renvoie le nombre spécifié de caractères situés le plus à gauche dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `position`— Position du caractère par laquelle commencer, à partir de l'extrémité gauche de la chaîne.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "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

Renvoie dans une nouvelle colonne la longueur des chaînes de la colonne source ou des chaînes personnalisées.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
```

```
        }
    }
}
```

LOWER

Convertit tous les caractères alphabétiques des chaînes de la colonne source ou des chaînes personnalisées en minuscules, et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
```

```
    }
}
```

FUSIONNER_COLONNES_ET_VALEURS

Concatène les chaînes des colonnes source et renvoie le résultat dans une nouvelle colonne. Vous pouvez insérer un délimiteur entre les valeurs fusionnées.

Parameters

- `sourceColumns`— Les noms d'au moins deux colonnes existantes, au JSON-encoded format.
- `delimiter` : facultatif. Un ou plusieurs caractères à placer entre les deux valeurs de colonne source.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
    }
}
```

CORRECT

Convertit tous les caractères alphabétiques des chaînes de la colonne source ou des valeurs personnalisées en majuscules et renvoie le résultat dans une nouvelle colonne.

Dans le cas approprié, également appelé majuscule, la première lettre de chaque mot est en majuscule et le reste du mot est transformé en minuscule. Un exemple est : Le renard brun rapide a sauté par-dessus la clôture

Parameters

- `sourceColumn` : nom d’une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_proper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "value": "MR. H. SMITH, ESQ.",
            "targetColumn": "formal_name_proper"
        }
    }
}
```

SUPPRIMER_SYMBOLES

Supprime les caractères autres que des lettres, des chiffres, des caractères latins accentués ou des espaces blancs des chaînes de la colonne source ou des chaînes personnalisées, et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "sourceColumn": "info_url",
            "targetColumn": "info_url_remove_symbols"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "value": "$&#$&HEY!#@@",
            "targetColumn": "without_symbols"
        }
    }
}
```

SUPPRIMER_WHITESPACE

Supprime les espaces blancs des chaînes de la colonne source ou des chaînes personnalisées, et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

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

CHAÎNE DE RÉPÉTITION

Répète les chaînes de la colonne source ou de la valeur d'entrée personnalisée un nombre de fois spécifié et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d’une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `count`— Le nombre de fois que la chaîne doit être répétée.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "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

À partir d'un certain nombre de caractères, prend le nombre de caractères le plus à droite dans les chaînes de la colonne source ou des chaînes personnalisées, et renvoie le nombre spécifié de caractères situés le plus à droite dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `position`— Position initiale du caractère, à partir du côté droit de la chaîne.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "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

En effectuant une recherche de droite à gauche, vous trouvez les chaînes correspondant à une chaîne spécifiée dans la colonne source ou à partir d'une valeur personnalisée, puis renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `pattern`— Expression régulière à rechercher.
- `position`— Position du caractère par laquelle commencer, à partir de l'extrémité droite de la chaîne.
- `ignoreCase`— Si`true`, ignorez les différences de majuscules (entre majuscules et minuscules) entre les lettres. Pour appliquer une correspondance stricte, utilisez `false` plutôt.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

Renvoie `true` une nouvelle colonne si un nombre spécifié de caractères situés le plus à gauche, ou une chaîne personnalisée, correspond à un modèle.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `pattern`— Expression régulière qui doit correspondre au début de la chaîne.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "STARTS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
        }
    }
}
```

STRING SUPÉRIEUR À

Crée une nouvelle colonne contenant l'un des éléments suivants :

- `True`si une chaîne d'une colonne (ou valeur) est supérieure à une autre chaîne d'une autre colonne (ou valeur).
- `False`s'il n'y a pas de correspondance.

Parameters

- `sourceColumn1` : nom d’une colonne existante.
- `sourceColumn2` : nom d’une colonne existante.
- `value1` : chaîne de caractères à évaluer.
- `value2` : chaîne de caractères à évaluer.

- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous ne pouvez spécifier qu'une seule des combinaisons suivantes :
>
> - Les deux`sourceColumn`*N*.
> - L'un des `sourceColumn`*N* et l'un des`value`*N*.
> - Les deux`value`*N*.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
        }
    }
}
```

STRING_GREATER_THAN_EQUAL

Crée une nouvelle colonne contenant l'un des éléments suivants :

- `True`si une chaîne d'une colonne (ou valeur) est supérieure ou égale à une autre chaîne d'une autre colonne (ou valeur).
- `False`s'il n'y a pas de correspondance.

Parameters

- `sourceColumn1` : nom d’une colonne existante.
- `sourceColumn2` : nom d’une colonne existante.
- `value1` : chaîne de caractères à évaluer.

- `value2` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous ne pouvez spécifier qu'une seule des combinaisons suivantes :
>
> - Les deux`sourceColumn`*N*.
> - L'un des `sourceColumn`*N* et l'un des`value`*N*.
> - Les deux`value`*N*.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "nationality",
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

CHAÎNE INFÉRIEURE À

Crée une nouvelle colonne contenant l'un des éléments suivants :

- `True`si une chaîne d'une colonne (ou valeur) est inférieure à une autre chaîne d'une autre colonne (ou valeur).
- `False`s'il n'y a pas de correspondance.

Parameters

- `sourceColumn1` : nom d’une colonne existante.
- `sourceColumn2` : nom d’une colonne existante.

- `value1` : chaîne de caractères à évaluer.
- `value2` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous ne pouvez spécifier qu'une seule des combinaisons suivantes :
>
> - Les deux`sourceColumn`*N*.
> - L'un des `sourceColumn`*N* et l'un des`value`*N*.
> - Les deux`value`*N*.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

CHAÎNE INFÉRIEURE À ÉGALE

Crée une nouvelle colonne contenant l'un des éléments suivants :

- `True`si une chaîne d'une colonne (ou valeur) est inférieure ou égale à une autre chaîne d'une autre colonne (ou valeur).
- `False`s'il n'y a pas de correspondance.

Parameters

- `sourceColumn1` : nom d’une colonne existante.

- `sourceColumn2` : nom d'une colonne existante.
- `value1` : chaîne de caractères à évaluer.
- `value2` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous ne pouvez spécifier qu'une seule des combinaisons suivantes :
>
> - Les deux`sourceColumn`*N*.
> - L'un des `sourceColumn`*N* et l'un des`value`*N*.
> - Les deux`value`*N*.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

Renvoie dans une nouvelle colonne certaines ou toutes les chaînes spécifiées dans la colonne source, en fonction des valeurs d'index de début et de fin définies par l'utilisateur.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `startPosition`— Position du caractère par laquelle commencer, à partir de l'extrémité gauche de la chaîne.

- `endPosition`— Position du caractère à terminer, à partir de l'extrémité gauche de la chaîne.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
}
```

TRIM

Supprime les espaces blancs de début et de fin des chaînes de la colonne source ou des chaînes personnalisées, et renvoie le résultat dans une nouvelle colonne. Les espaces entre les mots ne sont pas supprimés.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
    }
}
```

UNICODE

Renvoie dans une nouvelle colonne la valeur d'index Unicode pour le premier caractère des chaînes de la colonne source ou pour les chaînes personnalisées.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
        }
    }
}
```

UPPER

Convertit tous les caractères alphabétiques des chaînes de la colonne source ou des chaînes personnalisées en majuscules et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
        }
    }
}
```

Fonctions de date et d’heure

Vous trouverez ci-dessous des rubriques de référence pour les fonctions de date et d'heure qui fonctionnent avec les actions de recette.

Rubriques

- CONVERT_TIMEZONE
- DATE
- DATE_ADD
- DATE_DIFF
- FORMAT DE DATE
- DATE_HEURE
- DAY

- HOUR
- MILLISECOND
- MINUTE
- MONTH
- NOM_MOIS
- NOW
- TRIMESTRE
- SECOND
- TIME
- AUJOURD'HUI
- UNIX_TIME
- UNIX_TIME_FORMAT
- JOUR_SEMAINE
- NUMÉRO_SEMAINE
- YEAR

CONVERT_TIMEZONE

Convertit une valeur horaire de la colonne source en une nouvelle colonne basée sur un fuseau horaire spécifié.

Parameters

- `sourceColumn` : nom d'une colonne existante. La colonne source peut être de type `stringdate`, `outimestamp`.
- `fromTimeZone`— Fuseau horaire de la valeur source. Si rien n'est spécifié, le fuseau horaire par défaut est UTC.
- `toTimeZone`— Fuseau horaire à convertir. Si rien n'est spécifié, le fuseau horaire par défaut est UTC.
- `targetColumn`— Nom de la colonne nouvellement créée.
- `dateTimeFormat` : facultatif. Chaîne de format pour la date. Si le format n'est pas spécifié, le format par défaut est utilisé :`yyyy-mm-dd HH:MM:SS`.

Example Exemple

```
{
    "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

Crée une nouvelle colonne contenant la valeur de date, à partir des colonnes source ou des valeurs fournies.

Parameters

- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si cette chaîne n'est pas spécifiée, le format par défaut est`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Une JSON-encoded chaîne représentant les composants de la date et de l'heure :
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Chaque composant doit spécifier l'une des options suivantes :
  - `sourceColumn` : nom d’une colonne existante.
  - `value` : chaîne de caractères à évaluer.

- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "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

Ajoute une année, un mois ou un jour à la date à partir d'une colonne ou d'une valeur source, et crée une nouvelle colonne contenant les résultats.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `units`— Unité de mesure pour ajuster la date. Les valeurs valides sont `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, et`MINUTES`.
- `dateAddValue`— Le nombre de `units` à ajouter à la date.
- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si aucune valeur n'est spécifiée, le format par défaut est `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
}
```

DATE_DIFF

Crée une nouvelle colonne contenant la différence entre deux dates.

Parameters

- `sourceColumn1` : nom d'une colonne existante.
- `sourceColumn2` : nom d'une colonne existante.
- `value1` : chaîne de caractères à évaluer.
- `value2` : chaîne de caractères à évaluer.
- `units`— Unité de mesure pour décrire la différence entre les dates. Les valeurs valides sont `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, et`MINUTES`.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous ne pouvez spécifier que l'une des combinaisons suivantes :

- Les deux `sourceColumn1` et`sourceColumn2`.
- L'un des `sourceColumn1` ou `sourceColumn2` et l'un des `value1` ou`value2`.
- Les deux `value1` et`value2`.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
        }
    }
}
```

FORMAT DE DATE

Crée une nouvelle colonne contenant une date, dans un format spécifique, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value`— Chaîne à évaluer.
- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si aucune valeur n'est spécifiée, le format par défaut est `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemples

```
{
    "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_HEURE

Crée une nouvelle colonne contenant la valeur de date et d'heure, à partir des colonnes source ou des valeurs fournies.

Parameters

- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si cette chaîne n'est pas spécifiée, le format par défaut est`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Une JSON-encoded chaîne représentant les composants de la date et de l'heure :
  - `year`
  - `value`
  - `month`

- `day`
- `hour`
- `second`

Chaque composant doit spécifier l'une des options suivantes :

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.

Example Exemple

```
{
    "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

Crée une nouvelle colonne contenant le jour du mois, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

HOUR

Crée une nouvelle colonne contenant la valeur de l'heure, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
```

```
}
```

MILLISECOND

Crée une nouvelle colonne contenant la valeur en millisecondes d'une colonne source ou d'une valeur d'entrée.

Parameters

- `sourceColumn` : nom d'une colonne existante. La colonne source peut être de type `stringdate`, `outimestamp`.
- `value` : chaîne de caractères à évaluer.
- `targetColumn`— Nom de la colonne nouvellement créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
}
```

MINUTE

Crée une nouvelle colonne contenant la valeur des minutes, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.

- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
}
```

MONTH

Crée une nouvelle colonne contenant le numéro du mois, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

NOM_MOIS

Crée une nouvelle colonne contenant le nom du mois, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

NOW

Crée une nouvelle colonne contenant la date et l'heure actuelles au format`yyyy-mm-dd HH:MM:SS`.

Parameters

- `timeZone`— Le nom d'un fuseau horaire. Si aucun fuseau horaire n'est spécifié, la valeur par défaut est le temps universel coordonné (UTC).
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
}
```

TRIMESTRE

Crée une nouvelle colonne contenant le trimestre basé sur la date à partir d'une chaîne représentant une date.

> **Note**
>
> Les trimestres sont désignés dans la nouvelle colonne par 1, 2, 3 ou 4.
>
> - 1 correspond aux mois de janvier, février et mars.
> - 2 correspond aux mois d'avril, mai et juin.
> - Le 3 correspond aux mois de juillet, août et septembre.
> - Le 4 correspond aux mois d'octobre, de novembre et de décembre.

Parameters

- `sourceColumn` : nom d'une colonne existante. La colonne source peut être de type `stringdate`, `outimestamp`.
- `value` : chaîne de caractères à évaluer.
- `targetColumn`— Nom de la colonne nouvellement créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
}
```

SECOND

Crée une nouvelle colonne contenant la deuxième valeur, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

Crée une nouvelle colonne contenant la valeur temporelle, à partir des colonnes source ou des valeurs fournies.

Parameters

- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si cette chaîne n'est pas spécifiée, le format par défaut est`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Une JSON-encoded chaîne représentant les composants de la date et de l'heure :
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Chaque composant doit spécifier l'une des options suivantes :

  - `sourceColumn` : nom d'une colonne existante.
  - `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

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

AUJOURD'HUI

Crée une nouvelle colonne contenant la date actuelle au format`yyyy-mm-dd`.

Parameters

- `timeZone`— Le nom d'un fuseau horaire. Si aucun fuseau horaire n'est spécifié, la valeur par défaut est le temps universel coordonné (UTC).
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
```

```
        }
    }
}
```

UNIX_TIME

Crée une nouvelle colonne contenant un nombre représentant l'époque (heure Unix), c'est-à-dire le nombre de secondes écoulées depuis le 1er janvier 1970, en fonction d'une colonne source ou d'une valeur d'entrée. Si le fuseau horaire peut être déduit, la sortie correspond à ce fuseau horaire. Dans le cas contraire, la sortie est en temps universel coordonné (UTC).

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
        }
    }
}
```

UNIX_TIME_FORMAT

Convertit l'heure Unix d'une colonne source ou d'une valeur d'entrée en un format de date numérique spécifié et renvoie le résultat dans une nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value`— Un entier qui représente un horodatage d'une époque Unix.
- `dateTimeFormat` : facultatif. Chaîne de format pour la date, telle qu'elle doit apparaître dans la nouvelle colonne. Si aucune valeur n'est spécifiée, le format par défaut est `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

JOUR_SEMAINE

Crée une nouvelle colonne contenant le jour de la semaine, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

NUMÉRO_SEMAINE

Crée une nouvelle colonne contenant le numéro de la semaine (de 1 à 52), à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
```

```
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

YEAR

Crée une nouvelle colonne contenant l'année, à partir d'une chaîne représentant une date.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : nom de la colonne qui vient d'être créée.

> **Note**
>
> Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
}
```

Fonctions de fenêtrage

Vous trouverez ci-dessous des rubriques de référence pour les fonctions de fenêtre qui fonctionnent avec des actions de recette.

Rubriques

FILL

Renvoie une nouvelle colonne basée sur une colonne source spécifiée. Pour toute valeur manquante ou nulle dans la colonne source, `FILL` choisit la valeur non vide la plus récente dans une fenêtre de lignes avant et après la valeur source en question. La valeur choisie est ensuite placée dans la nouvelle colonne.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
}
```

NEXT

Renvoie une nouvelle colonne, où chaque valeur représente une valeur située n lignes plus loin dans la colonne source.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- numRows— Une valeur qui représente n lignes plus tôt dans la colonne source. Par exemple, si la valeur numRows est 3, NEXT utilise la troisième `sourceColumn` valeur suivante comme nouvelle `targetColumn` valeur.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
```

```
}
```

PRÉCÉDENT

Renvoie une nouvelle colonne, où chaque valeur représente une valeur située n lignes plus tôt dans la colonne source.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRows`— Une valeur qui représente n lignes plus tôt dans la colonne source. Par exemple, si la valeur `numRows` est 3, elle `PREV` utilise la troisième `sourceColumn` valeur précédente comme nouvelle `targetColumn` valeur.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

MOYENNE CONTINUE

Renvoie dans une nouvelle colonne la moyenne mobile des valeurs d'un nombre spécifié de lignes antérieures à un nombre spécifié de lignes situées après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.

- numRowsAfter— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- targetColumn : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
    }
}
```

ROLLING_COUNT_A

Renvoie dans une nouvelle colonne le nombre cumulé de valeurs non nulles entre un nombre spécifié de lignes antérieures et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- sourceColumn : nom d'une colonne existante.
- numRowsBefore— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- numRowsAfter— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- targetColumn : nom de la colonne qui vient d'être créée.

Example Exemple

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

```
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

Renvoie dans une nouvelle colonne la k ème plus grande valeur aléatoire entre un nombre spécifié de lignes avant et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `value`— La valeur de k.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
  "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

Renvoie dans une nouvelle colonne la k ème plus grande valeur unique mobile entre un nombre spécifié de lignes antérieures à un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `value`— La valeur de k.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
  "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

Renvoie dans une nouvelle colonne le maximum cumulatif de valeurs entre un nombre spécifié de lignes antérieures et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.

numRowsBefore— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.

- numRowsAfter— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- targetColumn : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_MAX",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

ROLLING_MIN

Renvoie dans une nouvelle colonne le minimum progressif de valeurs entre un nombre spécifié de lignes antérieures et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- sourceColumn : nom d'une colonne existante.

  numRowsBefore— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- numRowsAfter— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- targetColumn : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

MODE ROULANT

Renvoie dans une nouvelle colonne le mode de rotation (valeur la plus courante) d'un nombre spécifié de lignes antérieures à un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- numRowsBefore— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- numRowsAfter— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- ModeType — Fonction modale à appliquer à la fenêtre. Les valeurs valides sont `NONE`, `MINIMUM`, `MAXIMUM` et `AVERAGE`.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
```

```
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

ROLLING_STANDARD_DEVIATION

Renvoie dans une nouvelle colonne l'écart type progressif des valeurs entre un nombre spécifié de lignes avant et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

ROLLING_SUM

Renvoie dans une nouvelle colonne la somme progressive des valeurs d'un nombre spécifié de lignes antérieures à un nombre spécifié de lignes situées après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.

  `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

VARIANCE_VARIABLE

Renvoie dans une nouvelle colonne la variance progressive des valeurs entre un nombre spécifié de lignes avant et un nombre spécifié de lignes après la ligne actuelle dans la colonne spécifiée.

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `numRowsBefore`— Nombre de lignes avant la ligne source actuelle, représentant le début de la fenêtre.
- `numRowsAfter`— Nombre de lignes après la ligne source actuelle, représentant la fin de la fenêtre.
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

Renvoie dans une nouvelle colonne un identifiant de session basé sur une fenêtre créée par des noms de colonnes à partir des instructions « grouper par » et « trier par ».

Parameters

- `groupByColumns`— JSON-encoded Chaîne décrivant les colonnes « grouper par ».
- `orderByColumns`— JSON-encoded Chaîne décrivant les colonnes « trier par ».
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESSION

Renvoie dans une nouvelle colonne un identifiant de session basé sur une fenêtre créée par des noms de colonnes à partir des instructions « grouper par » et « trier par ».

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `units`— Une unité de mesure pour décrire la durée de la session. Les valeurs valides sont `MONTHSYEARS`, `MILLISECONDSQUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, et`MINUTES`.
- `value`— Le nombre de `units` pour définir la période.
- `groupByColumns`— JSON-encoded Chaîne décrivant les colonnes « grouper par ».
- `orderByColumns`— JSON-encoded Chaîne décrivant les colonnes « trier par ».
- `targetColumn` : nom de la colonne qui vient d'être créée.

Example Exemple

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
}
```

Fonctions Web

Vous trouverez ci-dessous des rubriques de référence pour les fonctions Web qui fonctionnent avec des actions de recette.

Rubriques

IP_TO_INT

Convertit la valeur IPv4 (Internet Protocol version 4) de la colonne source ou d'une autre valeur en valeur entière correspondante dans la colonne cible et renvoie le résultat dans une nouvelle colonne. Cette fonction ne fonctionne que pour IPv4.

Par exemple, considérez l'adresse IP suivante.

```
192.168.1.1
```

Si vous utilisez cette valeur comme entrée pour`IP_TO_INT`, la valeur de sortie est la suivante.

```
3232235777
```

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_TO_IP

Convertit la valeur entière de la colonne source ou d'une autre valeur en valeur IPv4 correspondante dans la colonne cible puis renvoie le résultat dans une nouvelle colonne. Cette fonction ne fonctionne que pour IPv4.

Par exemple, considérez le nombre entier suivant.

```
167772410
```

Si vous utilisez cette valeur comme entrée pour`INT_TO_IP`, la valeur de sortie est la suivante.

```
10.0.0.250
```

Parameters

- `sourceColumn` : nom d’une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
[  {
    "RecipeAction": {
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
  }
]
```

URL_PARAMS

Extrait les paramètres de requête d'une chaîne URL, les met en forme sous forme d'objet JSON et renvoie le résultat dans une nouvelle colonne.

Par exemple, considérez l'URL suivante.

```
https://example.com/?firstParam=answer&secondParam=42
```

Si vous utilisez cette valeur comme entrée pourURL_PARAMS, la valeur de sortie est la suivante.

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parameters

- `sourceColumn` : nom d'une colonne existante.
- `value` : chaîne de caractères à évaluer.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Vous pouvez préciser `sourceColumn` ou `value`, mais pas les deux.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
    }
}
```

Autres fonctions

Vous trouverez ci-dessous des rubriques de référence pour d'autres fonctions qui fonctionnent avec des actions de recette.

Rubriques

- COALESCE
- GET_ACTION_RESULT
- GET_STEP_DATAFRAME

COALESCE

Renvoie dans une nouvelle colonne la première valeur non nulle trouvée dans le tableau de colonnes. L'ordre des colonnes répertoriées dans la fonction détermine l'ordre dans lequel elles sont recherchées.

Parameters

- `sourceColumns`— JSON-encoded Chaîne représentant la liste des colonnes existantes.
- `targetColumn` : le nom de la nouvelle colonne à créer.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

Récupère le résultat d'une action précédemment soumise. À utiliser uniquement dans le cadre de l'expérience interactive.

Parameters

- `actionId`— Le ActionId renvoyé dans la SendProjectSessionAction réponse initiale.

Example Exemple

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

```
            "actionId": "7",
        }
    }
}
```

GET_STEP_DATAFRAME

Récupère le bloc de données à partir d'une étape de la recette du projet. À utiliser uniquement dans le cadre de l'expérience interactive. Utilisé avec le ViewFrame paramètre pour paginer dans un bloc de données volumineux.

Parameters

- `stepIndex`— L'index de l'étape de la recette du projet pour laquelle récupérer le bloc de données.

Example Exemple

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

Quotas pour AWS Glue DataBrew

Vous pouvez consulter vos quotas DataBrew de service dans la console AWS Service Quotas. Vous pouvez également demander une augmentation de quota, pour tout quota ajustable.

Historique du document pour AWS Glue DataBrew Guide du développeur

Version actuelle de l'API : databrew-2017-07-25

Le tableau suivant décrit la documentation de cette version de AWS Glue DataBrew. Si vous souhaitez être averti lorsque le guide du AWS Glue DataBrew développeur est mis à jour, vous pouvez vous abonner au flux RSS.

Modification	Description	Date
`glue:GetCustomEntityType` ajouté aux politiques AWS gérées	Cette autorisation est requise pour exécuter des tâches AWS Glue DataBrew de profil lorsque PII-identification cette option est activée. Pour plus d'informations, voir les AWS Glue DataBrew mises à jour des politiques AWS gérées.	20 mars 2024
Support de plusieurs algorithmes de hachage dans la transformation CRYPTOGRAPHIC_HASH	Vous pouvez désormais spécifier un algorithme de hachage lors du hachage de valeurs dans une colonne. Pour plus d'informations, consultez CRYPTOGRAPHIC_HASH.	11 août 2023
`glue:BatchGetCustomEntityTypes` ajouté aux politiques AWS gérées	Cette autorisation est requise pour exécuter des tâches AWS Glue DataBrew de profil lorsque PII-identification cette option est activée. Pour plus d'informations, voir les AWS Glue DataBrew mises à jour des politiques AWS gérées.	9 mai 2022

Support du format de fichier Apache ORC	DataBrew supporte désormais Apache ORC en tant que format de fichier pour les sources de DataBrew données et les sorties. Pour plus d'informations, consultez la section Types de fichiers pris en charge pour les sources de données.	31 mars 2022
Support pour l'accès entre comptes AWS Glue Data Catalog Amazon S3	Vous pouvez désormais accéder aux tables AWS Glue Data Catalog S3 à partir d'autres tables Comptes AWS si une politique de ressource s appropriée est créée dans la AWS Glue console. Après avoir créé une politique, les tables du catalogue de données S3 pertinentes peuvent être sélectionnées comme sources d'entrée lors de la création d'un DataBrew ensemble de données. Pour plus d'informations, consultez la section Connexions prises en charge pour les sources de données et les sorties.	11 mars 2022

Support pour l'intégration native de la console avec Amazon AppFlow	DataBrew intègre désormais une console native à Amazon AppFlow. Cette intégration signifie que vous pouvez vous connecter aux données de Salesforce, Zendesk, Slack et d'autres applications SaaS (Software-as-a-Service). ServiceNow Vous pouvez également vous connecter à des données provenant Services AWS notamment d'Amazon S3 et d'Amazon Redshift. Pour plus d'informations, consultez la section Connexions prises en charge pour les sources de données et les sorties.	18 novembre 2021
Support pour les règles de qualité des données	DataBrew prend désormais en charge la création de règles de qualité des données, qui sont des contrôles de validation personnalisables qui définissent les exigences commerciales pour des données spécifiques. Pour plus d'informations, consultez la section Validation de la qualité des données dans AWS Glue DataBrew.	18 novembre 2021

Support pour les instructions SQL personnalisées	DataBrew prend désormais en charge les instructions SQL personnalisées pour récupérer des données depuis Amazon Redshift et Snowflake . Cette prise en charge signifie que vous pouvez utiliser une requête spécialement conçue pour sélectionner et limiter les données renvoyées par de grandes tables. Pour plus d'informations, consultez la section Connexions prises en charge pour les sources de données et les sorties.	18 novembre 2021
Support pour la détection des informations personnelles	DataBrew prend désormais en charge la détection des informations personnel les identifiables (PII). Cela vous donne la possibilité de masquer les informati ons personnelles lors de la préparation des données. Pour plus d'informations, voir Identification et traitement des informations personnelles identifiables (PII).	18 novembre 2021
Support pour d'autres AWS régions	DataBrew prend désormais en charge des AWS régions supplémentaires. Pour obtenir la liste des régions prises en charge, consultez la section AWS Glue DataBrew Points de terminaison et quotas.	5 octobre 2021

Support pour l'écriture de données dans les tables Lake Formation-based Amazon S3	DataBrew prend désormais en charge l'écriture de données dans des tables AWS Glue Data Catalog S3 basées sur AWS Lake Formation. DataBrew prend également désormais en charge l'écriture de données au format Tableau Hyper. Pour plus d'informations, consultez la section Création et utilisation de tâches de AWS Glue DataBrew recette.	13 août 2021
Support pour l'écriture de données dans des destinations JDBC	DataBrew prend désormais en charge l'écriture de données directement dans les JDBC-supported bases de données et les entrepôts de données. Il s'agit notamment d'Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database et PostgreSQL. Pour plus d'informations, consultez la section Création et utilisation de tâches de AWS Glue DataBrew recette.	23 juillet 2021

Support pour spécifier quelles statistiques de qualité des données sont générées pour une tâche de profilage	DataBrew permet désormais de spécifier quelles statistiques de qualité des données sont générées automatiquement pour les ensembles de données dans une tâche de profilage. Pour plus d'informations, consultez la section Création et utilisation de tâches de AWS Glue DataBrew recette.	23 juillet 2021
Support pour l'écriture de jeux de données dans AWS Glue Data Catalog	DataBrew inclut désormais la prise en charge de l'écriture de jeux de données directement dans le AWS Glue Data Catalog. Vous pouvez choisir de stocker les ensembles de données créés à partir de tâches qui exécutent vos recettes de préparation des données dans les tables Amazon S3, Amazon Redshift et Amazon RDS du catalogue de données. Les tables RDS prises en charge incluent celles d'Amazon Aurora, RDS pour Oracle, RDS pour Microsoft SQL Server, RDS pour MySQL et RDS pour PostgreSQL.	30 Juin 2021

Support pour l'identification des types de données avancés	DataBrew inclut désormais la prise en charge de l'identif ication et du marquage automatiques des types de données avancés pour les colonnes, ce qui facilite la normalisation des colonnes contenant certains types de données. Ces types de données incluent le numéro de sécurité sociale, l'adresse e-mail, le numéro de téléphone , le sexe, la carte de crédit, l'URL, l'adresse IP, la date et l'heure, la devise, le code postal, le pays, la région, l'État et la ville.	30 Juin 2021
Support pour l'utilisation d'Amazon AppFlow pour transférer des données depuis des applications SAAS	DataBrew prend désormais en charge l'utilisation AppFlow d'Amazon pour transférer des données vers Amazon S3 à partir d'applications logicielles en tant que service (SaaS) tierces telles que Salesforce, Zendesk, Slack et. ServiceNow Pour plus d'informations, consultez la section Connexions prises en charge pour les sources de données et les sorties.	29 avril 2021

Support pour la création de DataBrew jeux de données avec des entrées provenant de bases de données JDBC	DataBrew permet désormais de créer des ensembles de données à partir de données contenues dans des JDBC-supported bases de données et des entrepôts de données, notamment Amazon Redshift, Snowflake , Microsoft SQL Server, MySQL, Oracle Database et PostgreSQL. Pour plus d'informations, consultez la section Connexions prises en charge pour les sources de données et les sorties.	2 avril 2021
Support pour des Régions AWS	DataBrew prend désormais en charge les options supplémen taires Régions AWS. Pour obtenir la liste des régions prises en charge, consultez la section AWS Glue DataBrew Points de terminaison et quotas.	28 janvier 2021

Nouvelles transformations pour gérer la duplication	Quatre nouvelles transformations pour gérer la duplication ont été ajoutées à la DataBrew console et à l'API. Pour plus d'informations, consultez DELETE_DUPLICATE_ROWS, FLAG_DUPLICATE_ROWS, FLAG_DUPLICATES_IN_COLUMNet REMOVE_DUPLICATES dans les étapes de la recette de qualité des données.	28 janvier 2021
Délimiteurs CSV supplémentaires	DataBrew prend désormais en charge des délimiteurs supplémentaires en plus des virgules dans les fichiers de valeurs séparées par des virgules (CSV) utilisés pour créer des ensembles de données. DataBrew Pour plus d'informations, consultez la section Création et utilisation de AWS Glue DataBrew jeux de données.	28 janvier 2021
DataBrew extension pour JupyterLab	Vous pouvez maintenant l'utiliser AWS Glue DataBrew comme extension dans JupyterLab. Pour plus d'informations, consultez la section Utilisation en DataBrew tant qu'extension dans JupyterLab.	20 novembre 2020

Nouvel outil de préparation des données :AWS Glue DataBrew	Il s’agit de la première version du Guide du développeur AWS Glue DataBrew.	11 novembre 2020

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.