aws

Unable to locate subtitle

AWS Glue DataBrew Entwicklerhandbuch

AWS Glue DataBrew Entwicklerhandbuch: ***Unable to locate subtitle***

Table of Contents

Was ist DataBrew? ... 1
Kernkonzepte und Begriffe ... 2
Projekte ... 3
Datensätze ... 3
Rezepte ... 4
Jobs ... 4
Datenherkunft ... 4
Datenprofil ... 4
Produkt- und Service-Integrationen ... 4
Einrichtung ... 8
Ein neues einrichten AWS Konto ... 8
Einrichtung der AWS CLI ... 10
Einrichten von IAM-Berechtigungen ... 11
Einrichtung von IAM-Richtlinien für DataBrew ... 13
Benutzer und Gruppen mit DataBrew Berechtigungen hinzufügen ... 26
Hinzufügen einer IAM-Rolle mit Berechtigungen DataBrew ... 27
Einrichtung AWS IAM Identity Center(IAM Identity Center) ... 28
Anmeldeschritte für einen IAM Identity-Benutzer Center-enabled ... 29
Benutzt DataBrew in JupyterLab ... 30
Voraussetzungen ... 31
Konfiguration JupyterLab für die Verwendung der Erweiterung ... 33
Aktivierung der Erweiterung für DataBrew JupyterLab ... 35
Erste Schritte ... 37
Voraussetzungen ... 37
Schritt 1: Erstellen eines Projekts ... 37
Schritt 2: Fassen Sie die Daten zusammen ... 38
Schritt 3: Weitere Transformationen hinzufügen ... 40
Schritt 4: Überprüfe deine DataBrew Ressourcen ... 41
Schritt 5: Erstellen Sie ein Datenprofil ... 41
Schritt 6: Transformieren Sie den Datensatz ... 43
Schritt 7: (Optional) Aufräumen ... 45
Datensätze ... 46
Unterstützte Dateitypen für Datenquellen ... 46
Unterstützte Verbindungen für Datenquellen und Ausgaben ... 48

Verwenden von Datensätzen ... 53
Löschen eines Datensatzes ... 57
Verbindung zu Ihren Daten herstellen ... 57
Verwenden von JDBC-Treibern zum Verbinden von Daten ... 58
Unterstützte JDBC-Treiber ... 60
Verbindung zu Daten in einer Textdatei herstellen mit DataBrew ... 61
Daten in mehreren Dateien in Amazon S3 verbinden ... 63
Schemas bei der Verwendung mehrerer Dateien als Datensatz ... 64
Verwenden von parametrisierten Pfaden für Amazon S3 ... 64
Datentypen ... 76
Fortgeschrittene Datentypen ... 77
Erweiterte Datentypen ... 77
Validierung der Datenqualität ... 79
Validierung der Datenqualitätsregeln ... 80
Auf der Grundlage der Validierungsergebnisse handeln ... 80
Einen Regelsatz mit Datenqualitätsregeln erstellen ... 81
Einen Profiljob erstellen ... 83
Überprüfung der Validierungsergebnisse und Aktualisierung der Datenqualitätsregeln ... 84
Verfügbare Schecks ... 85
Projekte ... 104
Erstellen eines Projekts ... 105
Überblick über eine DataBrew Projektsitzung ... 107
Rasteransicht ... 107
Schema-Ansicht ... 109
Profilansicht ... 111
Löschen eines Projekts ... 113
Rezepte ... 114
Veröffentlichung einer neuen Rezeptversion ... 115
Definition einer Rezeptstruktur ... 115
Verwenden von Bedingungen ... 119
Jobs ... 123
Jobs im Bereich Rezepte ... 123
Beispiel für die Partitionierung von Spalten ... 128
Automatisieren von Jobläufen mit einem Zeitplan ... 128
Mit Cron-Ausdrücken für Rezeptjobs arbeiten ... 130
Jobs und Jobpläne löschen ... 133

Stellen profilieren ... 134
Programmgesteuertes Erstellen einer Profiljobkonfiguration ... 136
Sicherheit ... 152
Datenschutz ... 153
Verschlüsselung im Ruhezustand ... 154
Verschlüsselung während der Übertragung ... 157
Schlüsselverwaltung ... 158
Identifizierung und Umgang mit personenbezogenen Daten ... 158
DataBrew Abhängigkeit von anderen AWS Dienstleistungen ... 159
Identity and Access Management ... 160
Authentifizierung mit Identitäten ... 160
Verwalten des Zugriffs mit Richtlinien ... 161
AWS Glue DataBrew und AWS Lake Formation ... 163
Wie AWS Glue DataBrew funktioniert mit IAM ... 164
Identity-based politische Beispiele ... 168
AWS Verwaltete Richtlinien für DataBrew ... 172
Fehlerbehebung ... 178
Protokollierung und Überwachung ... 180
Compliance-Validierung ... 181
Ausfallsicherheit ... 181
Sicherheit der Infrastruktur ... 182
Verwenden AWS Glue DataBrew mit deiner VPC ... 182
Verwenden AWS Glue DataBrew mit VPC-Endpunkten ... 183
Konfiguration und Schwachstellenanalyse in AWS Glue DataBrew ... 183
Überwachung DataBrew ... 184
Überwachung mit CloudWatch ... 185
Automatisieren mit Ereignissen CloudWatch ... 185
Überwachung mit CloudWatch Protokollen ... 188
Protokollierung von CloudTrail-API-Aufrufen mit ... 188
DataBrew Informationen in CloudTrail ... 188
Grundlegendes zu DataBrew Einträgen in Protokolldateien ... 189
Verwenden AWS Benutzerbenachrichtigungen mit AWS Glue Databrew ... 191
Referenz zu Rezeptschritt und Funktion ... 192
Grundlegende Schritte für Spaltenrezepte ... 194
CHANGE_DATA_TYPE ... 195
DELETE ... 196

DUPLIKAT ... 196
JSON_TO_STRUCTS ... 197
MOVE_AFTER ... 198
MOVE_BEFORE ... 198
MOVE_TO_END ... 199
MOVE_TO_INDEX ... 199
MOVE_TO_START ... 200
RENAME ... 200
SORT ... 201
TO_BOOLEAN_COLUMN ... 202
TO_DOUBLE_COLUMN ... 203
TO_NUMBER_COLUMN ... 204
TO_STRING_COLUMN ... 204
Rezeptschritte für die Datenbereinigung ... 205
CAPITAL_CASE ... 206
FORMAT_DATE ... 206
KLEINGESCHRIEBENES ... 207
GROSSBUCHSTABEN ... 208
SENTENCE_CASE ... 208
ADD_DOUBLE_QUOTES ... 209
ADD_PREFIX ... 209
ADD_SINGLE_QUOTES ... 210
ADD_SUFFIX ... 210
EXTRACT_BETWEEN DELIMITERS ... 211
EXTRACT_BETWEEN POSITIONS ... 211
EXTRACT_PATTERN ... 212
EXTRACT_VALUE ... 213
REMOVE_COMBINED ... 214
ERSETZEN_ZWISCHEN_TRENNZEICHEN ... 218
ERSETZEN_ZWISCHEN_POSITIONEN ... 218
REPLACE_TEXT ... 219
Rezeptschritte zur Datenqualität ... 220
ADVANCED_DATATYPE_FILTER ... 221
ADVANCED_DATATYPE_FLAG ... 223
DELETE_DUPLICATE_ROWS ... 224
EXTRACT_ADVANCED_DATATYPE_DETAILS ... 224

FILL_WITH_AVERAGE ... 225
FILL_WITH_CUSTOM ... 226
FILL_WITH_EMPTY ... 226
FILL_WITH_LAST_VALID ... 227
FILL_WITH_MEDIAN ... 228
FILL_WITH_MODE ... 228
FILL_WITH_MOST_FREQUENT ... 229
FILL_WITH_NULL ... 229
FILL_WITH_SUM ... 230
FLAG_DUPLICATE_ROWS ... 230
FLAG_DUPLICATES_IN_COLUMN ... 231
GET_ADVANCED_DATATYPE ... 232
REMOVE_DUPLICATES ... 232
REMOVE_INVALID ... 233
REMOVE_MISSING ... 233
REPLACE_WITH_AVERAGE ... 234
REPLACE_WITH_CUSTOM ... 234
ERSETZEN_DURCH_LEER ... 235
ERSETZEN_DURCH_LETZTE_VALIDE ... 236
ERSETZE DURCH_MEDIAN ... 237
ERSETZEN_MIT_MODUS ... 237
ERSETZEN_DURCH_MEISTES_HÄUFIGES_ ... 238
ERSETZEN_MIT_NULL ... 238
ERSETZE DURCH_ROLLING_AVERAGE ... 239
REPLACE_WITH_ROLLING_SUM ... 240
REPLACE_WITH_SUM ... 241
Schritte zur Rezeptur von PII ... 241
CRYPTOGRAPHIC_HASH ... 242
ENTSCHLÜSSELN ... 244
DETERMINISTIC_DECRYPT ... 245
DETERMINISTIC_ENCRYPT ... 246
VERSCHLÜSSELN ... 247
MASK_CUSTOM ... 249
MASK_DATE ... 250
MASK_DELIMITER ... 250
MASK_RANGE ... 251

ERSETZEN_MIT_RANDOM_BETWEEN ... 252
ERSETZEN_DURCH_ZUFÄLLIGE_DATE_BETWEEN ... 253
SHUFFLE_ROWS ... 254
Rezeptschritte zur Erkennung und Behandlung von Ausreißern ... 254
FLAGGENAUSREISSER ... 254
AUSREISSER ENTFERNEN ... 257
AUSREISSER ERSETZEN ... 259
AUSREISSER MIT Z-SCORE NEU SKALIEREN ... 262
AUSREISSER MIT SCHRÄGLAGE NEU SKALIEREN ... 264
Rezeptschritte für die Spaltenstruktur ... 266
BOOLESCHE_OPERATION ... 267
CASE_OPERATION ... 283
FLAG_COLUMN_FROM_NULL ... 296
FLAG_COLUMN_FROM_PATTERN ... 296
MERGE ... 297
SPLIT_COLUMN_BETWEEN_DELIMITER ... 298
SPLIT_COLUMN_BETWEEN_POSITIONS ... 299
SPLIT_COLUMN_FROM_END ... 299
SPLIT_COLUMN_FROM_START ... 300
SPLIT_COLUMN_MULTIPLE_DELIMITER ... 300
SPLIT_COLUMN_SINGLE_DELIMITER ... 301
SPLIT_COLUMN_WITH_INTERVALS ... 302
Rezeptschritte zur Spaltenformatierung ... 302
NUMBER_FORMAT ... 303
TELEFONNUMMER_FORMATIEREN ... 304
Rezeptschritte für die Datenstruktur ... 306
NEST_TO_ARRAY ... 306
NEST_TO_MAP ... 307
NEST_TO_STRUCT ... 308
UNNEST_ARRAY ... 309
UNNEST_MAP ... 309
UNNEST_STRUCT ... 310
UNNEST_STRUCT_N ... 311
GROUP_BY ... 312
JOIN ... 313
PIVOT ... 314

SCALE ... 315
TRANSPONIEREN ... 315
UNION ... 317
UNPIVOT ... 318
Rezeptschritte für Datenwissenschaft ... 318
BINARISIERUNG ... 319
BUCKETISIERUNG ... 320
CATEGORICAL_MAPPING ... 321
ONE_HOT_ENCODING ... 322
SCALE ... 315
SCHIEFHEIT ... 324
TOKENISIERUNG ... 325
Mathematische Funktionen ... 326
ABSOLUTE ... 327
ADD ... 327
CEILING ... 328
DEGREES ... 329
TEILEN ... 329
EXPONENT ... 330
FLOOR ... 330
IST_GERADE ... 331
IS_ODD ... 332
LN ... 332
LOG ... 333
MOD ... 334
MULTIPLIZIEREN ... 334
NEGIEREN ... 335
PI ... 335
POWER ... 336
RADIANS ... 337
RANDOM ... 337
RANDOM_BETWEEN ... 338
ROUND ... 338
SIGN ... 339
SQUARE_ROOT ... 339
SUBTRAHIEREN ... 340

Aggregationsfunktionen ... 341
ANY ... 341
AVERAGE ... 342
COUNT ... 342
COUNT_DISTINCT ... 343
KTH_LARGEST ... 344
KTH_LARGEST_UNIQUE ... 344
MAX ... 345
MEDIAN ... 346
MIN ... 346
MODE ... 347
STANDARD_DEVIATION ... 347
SUM ... 348
VARIANCE ... 349
Textfunktionen ... 349
CHAR ... 350
ENDS_WITH ... 351
EXAKT ... 352
FINDEN ... 353
LEFT ... 354
LEN ... 355
LOWER ... 356
MERGE_COLUMNS_AND_VALUES ... 357
RICHTIG ... 358
REMOVE_SYMBOLS ... 359
REMOVE_WHITESPACE ... 360
REPEAT_STRING ... 361
RIGHT ... 362
RIGHT_FIND ... 363
STARTS_WITH ... 364
STRING_GREATER_THAN ... 365
STRING_GREATER_THAN_EQUAL ... 366
STRING_LESS_THAN ... 367
STRING_LESS_THAN_EQUAL ... 368
SUBSTRING ... 369
TRIM ... 370

UNICODE ... 371
UPPER ... 372
Datums- und Zeitfunktionen ... 373
CONVERT_TIMEZONE ... 374
DATE ... 375
DATE_ADD ... 376
DATE_DIFF ... 377
DATUMSFORMAT ... 378
DATE_TIME ... 379
TAG ... 380
STUNDE ... 381
MILLISEKUNDE ... 381
MINUTE ... 382
MONAT ... 383
MONATSNAME ... 383
NOW ... 384
QUARTAL ... 385
SECOND ... 386
TIME ... 386
HEUTE ... 388
UNIX_TIME ... 388
UNIX_TIME_FORMAT ... 389
WOCHENTAG ... 390
WOCHE_NUMMER ... 391
JAHR ... 391
Fensterfunktionen ... 392
FILL ... 393
NEXT ... 394
ZURÜCK ... 394
ROLLING_AVERAGE ... 395
ROLLING_COUNT_A ... 396
ROLLING_KTH_LARGEST ... 396
ROLLING_KTH_LARGEST_UNIQUE ... 397
ROLLING_MAX ... 398
ROLLING_MIN ... 399
ROLLING_MODE ... 400

ROLLING_STANDARD_DEVIATION 400
ROLLING_SUM 401
ROLLING_VARIANCE 402
ROW_NUMBER 403
SESSION 403
Web-Funktionen 404
IP_TO_INT 405
INT_TO_IP 405
URL_PARAMS 406
Andere Funktionen 407
COALESCE 407
GET_ACTION_RESULT 408
GET_STEP_DATAFRAME 409
Kontingente und Beschränkungen 410
Dokumentverlauf 411
AWS Glossar 419
..... cdxx

Was ist AWS Glue DataBrew?

AWS Glue DataBrew ist ein visuelles Datenvorbereitungstool, mit dem Benutzer Daten bereinigen und normalisieren können, ohne Code schreiben zu müssen. Durch die Verwendung kann der Zeitaufwand für die Vorbereitung von Daten für Analysen und maschinelles Lernen (ML) im Vergleich zur kundenspezifisch entwickelten Datenaufbereitung um bis zu 80 Prozent DataBrew reduziert werden. Sie können aus über 250 vorgefertigten Transformationen wählen, um Datenaufbereitungsaufgaben wie das Filtern von Anomalien, das Konvertieren von Daten in Standardformate und das Korrigieren ungültiger Werte zu automatisieren.

Auf DataBrew diese Weise können Geschäftsanalysten, Datenwissenschaftler und Dateningenieure einfacher zusammenarbeiten, um Erkenntnisse aus Rohdaten zu gewinnen. Da DataBrew es serverlos ist, können Sie unabhängig von Ihrem technischen Niveau Terabyte an Rohdaten untersuchen und transformieren, ohne Cluster erstellen oder eine Infrastruktur verwalten zu müssen.

Mit der intuitiven DataBrew Benutzeroberfläche können Sie Rohdaten interaktiv entdecken, visualisieren, bereinigen und transformieren. DataBrew macht intelligente Vorschläge, um Ihnen bei der Identifizierung von Datenqualitätsproblemen zu helfen, die möglicherweise schwer zu finden und zeitaufwändig zu beheben sind. Bei der DataBrew Vorbereitung Ihrer Daten können Sie Ihre Zeit nutzen, um auf die Ergebnisse zu reagieren und schneller zu iterieren. Sie können die Transformation als Schritte in einem Rezept speichern, das Sie später aktualisieren oder mit anderen Datensätzen wiederverwenden und fortlaufend bereitstellen können.

Die folgende Abbildung zeigt, wie das auf hoher Ebene DataBrew funktioniert.

Zur Verwendung DataBrew erstellen Sie ein Projekt und stellen eine Verbindung zu Ihren Daten her. Im Projektarbeitsbereich werden Ihre Daten in einer rasterähnlichen visuellen Oberfläche angezeigt. Hier können Sie die Daten untersuchen und sich Wertverteilungen und Diagramme ansehen, um ihr Profil zu verstehen.

Um die Daten aufzubereiten, können Sie aus mehr als 250 Point-and-Click-Transformationen wählen. Dazu gehören das Entfernen von Nullen, das Ersetzen fehlender Werte, das Beheben von Schemainkonsistenzen, das Erstellen von Spalten auf der Grundlage von Funktionen und vieles mehr. Sie können Transformationen auch verwenden, um NLP-Techniken (Natural Language Processing) anzuwenden, um Sätze in Phrasen aufzuteilen. Sofortige Vorschauen zeigen einen Teil Ihrer Daten vor und nach der Transformation, sodass Sie Ihr Rezept ändern können, bevor Sie es auf den gesamten Datensatz anwenden.

Nachdem DataBrew Sie Ihr Rezept für Ihren Datensatz ausgeführt haben, wird die Ausgabe in Amazon Simple Storage Service (Amazon S3) gespeichert. Nachdem sich Ihr bereinigter, vorbereiteter Datensatz in Amazon S3 befindet, kann ihn ein anderes Ihrer Datenspeicher- oder Datenverwaltungssysteme aufnehmen.

Kernkonzepte und Begriffe in AWS Glue DataBrew

Im Folgenden finden Sie einen Überblick über die Kernkonzepte und die Terminologie unter AWS Glue DataBrew. Nachdem Sie diesen Abschnitt gelesen habenErste Schritte mit AWS

Glue DataBrew, finden Sie hier eine Anleitung zum Erstellen von Projekten, zum Verbinden von Datensätzen und zum Ausführen von Aufträgen.

Themen

- Projekt
- Datensatz
- Rezept
- Aufgabe
- Datenherkunft
- Datenprofil

Projekt

Der interaktive Arbeitsbereich für die Datenvorbereitung in DataBrew wird als Projekt bezeichnet. Mithilfe eines Datenprojekts verwalten Sie eine Sammlung verwandter Elemente: Daten, Transformationen und geplante Prozesse. Im Rahmen der Projekterstellung wählen oder erstellen Sie einen Datensatz, an dem Sie arbeiten möchten. Als Nächstes erstellen Sie ein Rezept, bei dem es sich um eine Reihe von Anweisungen oder Schritten handelt, DataBrew nach denen Sie handeln möchten. Durch diese Aktionen werden Ihre Rohdaten in ein Formular umgewandelt, das bereit ist, von Ihrer Datenpipeline verarbeitet zu werden.

Datensatz

Datensatz bedeutet einfach eine Reihe von Daten — Zeilen oder Datensätze, die in Spalten oder Felder unterteilt sind. Wenn Sie ein DataBrew Projekt erstellen, stellen Sie eine Verbindung zu Daten her, die Sie transformieren oder vorbereiten möchten, oder laden sie hoch. DataBrew kann mit Daten aus beliebigen Quellen arbeiten, die aus formatierten Dateien importiert wurden, und stellt eine direkte Verbindung zu einer wachsenden Liste von Datenspeichern her.

Denn DataBrew ein Datensatz ist eine schreibgeschützte Verbindung zu Ihren Daten. DataBrew sammelt eine Reihe von beschreibenden Metadaten, die auf die Daten verweisen. Keine tatsächlichen Daten können von geändert oder gespeichert werden DataBrew. Der Einfachheit halber verwenden wir Datensatz, um sowohl auf den tatsächlichen Datensatz als auch auf die DataBrew verwendeten Metadaten zu verweisen.

Rezept

Bei DataBrew einem Rezept handelt es sich um eine Reihe von Anweisungen oder Schritten für Daten, auf deren Grundlage Sie handeln DataBrew möchten. Ein Rezept kann viele Schritte enthalten, und jeder Schritt kann viele Aktionen enthalten. Sie verwenden die Transformationswerkzeuge auf der Werkzeugleiste, um alle Änderungen einzurichten, die Sie an Ihren Daten vornehmen möchten. Später, wenn Sie bereit sind, das fertige Produkt Ihres Rezepts zu sehen, weisen Sie diesen Job zu DataBrew und planen ihn. DataBrew speichert die Anweisungen zur Datentransformation, aber es werden keine Ihrer tatsächlichen Daten gespeichert. Sie können Rezepte herunterladen und in anderen Projekten wiederverwenden. Sie können auch mehrere Versionen eines Rezepts veröffentlichen.

Aufgabe

DataBrew übernimmt die Aufgabe, Ihre Daten zu transformieren, indem es die Anweisungen ausführt, die Sie bei der Erstellung eines Rezepts eingerichtet haben. Das Ausführen dieser Anweisungen wird als Job bezeichnet. Ein Job kann Ihre Datenrezepte nach einem voreingestellten Zeitplan in die Tat umsetzen. Sie sind jedoch nicht auf einen Zeitplan beschränkt. Sie können Jobs auch bei Bedarf ausführen. Wenn Sie einige Daten profilieren möchten, benötigen Sie kein Rezept. In diesem Fall können Sie einfach einen Profiljob einrichten, um ein Datenprofil zu erstellen.

Datenherkunft

DataBrew verfolgt Ihre Daten in einer visuellen Oberfläche, um deren Herkunft zu bestimmen, was als Datenherkunft bezeichnet wird. Diese Ansicht zeigt Ihnen, wie die Daten durch verschiedene Entitäten fließen, aus denen sie ursprünglich stammen. Sie können ihren Ursprung, andere Entitäten, von denen sie beeinflusst wurden, sehen, was mit den Daten im Laufe der Zeit passiert ist und wo sie gespeichert wurden.

Datenprofil

Wenn Sie ein Profil für Ihre Daten erstellen, DataBrew wird ein Bericht erstellt, der als Datenprofil bezeichnet wird. Diese Zusammenfassung informiert Sie über die aktuelle Form Ihrer Daten, einschließlich des Kontextes des Inhalts, der Struktur der Daten und ihrer Beziehungen. Sie können ein Datenprofil für jeden Datensatz erstellen, indem Sie einen Datenprofiljob ausführen.

Produkt- und Service-Integrationen

In diesem Abschnitt erfahren Sie, welche Produkte und Dienste integriert DataBrew werden können.

DataBrew funktioniert mit den folgenden AWS Diensten für Netzwerke, Verwaltung und Verwaltung:

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

DataBrew funktioniert mit den folgenden AWS Data Lakes und Datenspeichern:

- AWS Lake Formation
- Amazon S3

DataBrew unterstützt die folgenden Dateiformate und Erweiterungen für das Hochladen von Daten.

Format	Dateierweiterung (optional)	Erweiterungen für komprimierte Dateien (erforderlich)

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

Microsoft Excel-Arbeitsmappe	`.xlsx`	Keine Unterstützung für Komprimierung

JSON (JSON-Dokument und JSON-Zeilen)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`

Format	Dateierweiterung (optional)	Erweiterungen für komprimierte Dateien (erforderlich)
		`.deflate`
Apache ORC	`.orc`	`.zlib` `.snappy`

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

DataBrew schreibt Ausgabedateien in Amazon S3 und unterstützt die folgenden Dateiformate und Erweiterungen.

Format	Dateierweiterung (unkomprimiert)	Dateierweiterungen (komprimiert)
Comma-separated Werte	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated Werte	`.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 Parkett	Nicht unterstützt	`.glue.parquet.snappy`
Apache Avro	`.avro`	`.avro.snappy` , `.avro.gz`, `.avro.lz4` , `.avro.bz2`

Format	Dateierweiterung (unkomprimiert)	Dateierweiterungen (komprimiert)
		, `.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 (nur JSON-Zeilenformat)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Hyper Tableau	Nicht unterstützt	Nicht zutreffend

Einrichtung AWS Glue DataBrew

Bevor Sie beginnen AWS Glue DataBrew, müssen Sie einige Berechtigungen, einen Benutzer und eine Rolle einrichten. Führen Sie zunächst die folgenden Schritte aus:

1. Eröffnen Sie nach Bedarf ein AWS Konto und erstellen Sie AWS Identity and Access Management(IAM-) Richtlinien, damit Benutzer Folgendes ausführen DataBrew können:
   - Registrierung für ein neues AWS Konto und Hinzufügen eines Benutzers. Weitere Informationen finden Sie unter Ein neues einrichten AWS Konto.
   - Hinzufügen einer IAM-Richtlinie für einen Konsolenbenutzer. Ein Benutzer mit diesen Berechtigungen kann DataBrew auf die zugreifen AWS-Managementkonsole.
   - Hinzufügen von Berechtigungen für Datenressourcen für eine IAM-Rolle. Eine IAM-Rolle mit diesen Berechtigungen kann im Namen des Benutzers auf Daten zugreifen.

   Sie müssen ein IAM-Administrator sein, um Benutzer, Rollen und Richtlinien erstellen zu können.
2. Benutzer oder Gruppen hinzufügen für DataBrew. Ein Benutzer oder eine Gruppe mit den richtigen Berechtigungen kann DataBrew über die Konsole darauf zugreifen.
3. Hinzufügen einer Rolle mit Berechtigungen für den Datenzugriff DataBrew. Eine Rolle mit den richtigen Berechtigungen kann im Namen des Benutzers auf Daten zugreifen.

Ein neues einrichten AWS Konto

Wenn Sie noch kein AWS Konto haben, registrieren Sie sich für ein AWS Konto und erstellen Sie einen IAM-Administratorbenutzer.

Wenn Sie noch kein Konto haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Wählen Sie zum Erstellen eines Administratorbenutzers eine der folgenden Optionen aus.

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
Im IAM Identity Center (Empfohlen)	Verwendung von kurzfristigen Anmeldeinformationen für den Zugriff auf AWS. Dies steht im Einklang mit den bewährten Methoden für die Sicherheit. Weitere Informationen zu bewährten Methoden finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.	Beachtung der Anweisungen unter Erste Schritte im AWS IAM Identity Center-Benutzerhandbuch.	Konfigurieren Sie den programmatischen Zugriff, indem Sie AWS CLI die Konfiguration für die Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch vornehmen.

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
In IAM (Nicht empfohlen)	Verwendung von langfristigen Anmeldeinformationen für den Zugriff auf AWS.	Folgen Sie den Anleitungen unter IAM-Benutzer für den Notfallzugriff erstellen im IAM-Benutzerhandbuch.	Sie konfigurieren den programmgesteuerten Zugriff unter Verwendung der Informationen unter Verwalten der Zugriffsschlüssel für IAM-Benutzer im IAM-Benutzerhandbuch.

Weitere Informationen finden Sie unter folgenden Themen im IAM-Benutzerhandbuch:

- Was ist IAM?
- Erste Schritte mit IAM
- Erstellen eines Administrationsbenutzers und einer Administratorgruppe (Konsole)

Einrichtung der AWS CLI

Wenn Sie die DataBrew API verwenden JupyterLab möchten, stellen Sie sicher, dass Sie die AWS Command Line Interface(AWS CLI) installieren. Sie benötigen es nicht, um die DataBrew Konsole zu verwenden oder die Schritte in den Übungen Erste Schritte auszuführen.

Um das einzurichten AWS CLI

1. Laden Sie das herunter und konfigurieren Sie es AWS CLI mithilfe der folgenden Schritte:
   - Installieren des AWS CLI
   - Grundlagen der Konfiguration

2. Überprüfen Sie das Setup, indem Sie an der Eingabeaufforderung den folgenden DataBrew Befehl eingeben.

   ```
   aws databrew help
   ```

   Wenn diese Anweisung den Fehler "`aws: error: argument command: Invalid choice`" gefolgt von einer langen Liste von Diensten zurückgibt, deinstallieren Sie den AWS CLI und installieren Sie ihn anschließend erneut. Durch diese Aktion wird Ihre bestehende Konfiguration nicht überschrieben.

AWS CLI Befehle verwenden die AWS Standardregion aus Ihrer Konfiguration, sofern Sie sie nicht mit einem Parameter oder einem Profil festlegen. Sie können den `--region` Parameter zu jedem Befehl hinzufügen.

Wenn Sie möchten, können Sie ein benanntes Profil in `~/.aws/config` oder `%UserProfile %/.aws/config` (unter Microsoft Windows) hinzufügen. Benannte Profile können auch andere Einstellungen beibehalten, wie im folgenden Beispiel gezeigt.

```
[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
```

Einrichtung AWS Identity and Access Management(IAM) - Berechtigungen

Bevor Sie beginnen, müssen Sie einige Dinge in IAM einrichten. Sie müssen ein Administrator sein oder Hilfe von einem haben. Wenn Sie jedoch ein Konto mit Administratorzugriff haben, können Sie diese Aufgaben selbst erledigen. In diesem Abschnitt finden Sie einfache Anweisungen für jede Aufgabe.

Im Folgenden finden Sie eine Übersicht darüber, was Sie tun müssen:

- Im Rahmen dieses Vorgangs fügen Sie einen Benutzer hinzu. Sie müssen keinen neuen Benutzer hinzufügen, Sie können einen vorhandenen verwenden. Sie fügen DataBrew Berechtigungen hinzu, damit der Benutzer die DataBrew Konsole öffnen kann.

- Erstellen Sie eine IAM-Rolle. Eine Rolle erlaubt bestimmte Aktionen und erteilt innerhalb bestimmter Grenzen Berechtigungen, wenn sie verwendet wird. Sie funktioniert beispielsweise nur für Benutzer in Ihrem AWS Konto. Sie können später weitere Einschränkungen hinzufügen.
- Erstellen Sie die IAM-Richtlinie oder Richtlinien, die Sie benötigen. Eine Richtlinie ist eine Liste von Dingen, die ein Benutzer tun darf. Um eine Richtlinie zu erstellen, öffnen Sie eine weitere Konsolenseite und fügen den Text aus einer heruntergeladenen Datei ein.

> **Note**
>
> Was wir hier bereitstellen, sind grundlegende Einrichtungsinformationen. Wir empfehlen Ihnen, sich Zeit zu nehmen, um Ihre Berechtigungen so anzupassen, dass sie Ihren Sicherheits- und Compliance-Anforderungen entsprechen. Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren Administrator oder AWS Support.

Um die erforderlichen Berechtigungen hinzuzufügen

1. Gehen Sie wie folgt vor, um IAM-Richtlinien zu erstellen, damit Benutzer sie ausführen DataBrew können:

   - Fügen Sie eine benutzerdefinierte IAM-Richtlinie für einen Konsolenbenutzer hinzu. Wenn Sie keine benutzerdefinierte Richtlinie benötigen, können Sie stattdessen die AWS-verwaltete Richtlinie wählen. Fügen Sie sie einfach dem Benutzer in Schritt 2 hinzu. Ein Benutzer mit diesen Berechtigungen kann auf die DataBrew Servicekonsole zugreifen.
   - Fügen Sie Berechtigungen für Datenressourcen hinzu. Eine IAM-Rolle mit diesen Berechtigungen kann im Namen des Benutzers auf Daten zugreifen.

   Sie müssen Administrator sein, um Benutzer, Rollen und Richtlinien erstellen zu können.
2. Fügen Sie Benutzer oder Gruppen für hinzu DataBrew. Ein Benutzer oder eine Gruppe mit den richtigen Berechtigungen kann auf die DataBrew Konsole zugreifen.
3. Fügen Sie eine Rolle mit Zugriffsberechtigungen für Daten hinzu DataBrew. Eine Rolle mit den richtigen Berechtigungen kann im Namen des Benutzers auf Daten zugreifen.

Einrichtung von IAM-Richtlinien für DataBrew

Sie verwenden IAM-Richtlinien, um Berechtigungen zu verwalten. Eine Richtlinie macht es einfacher, zugehörige Berechtigungen auf einmal hinzuzufügen, anstatt sie einzeln hinzuzufügen.

Wir empfehlen, dass Sie die Richtlinien mit den gleichen Namen erstellen, die wir angeben. Wir verwenden in der gesamten Dokumentation die folgenden Namen für diese Richtlinien. Wenn Sie diese Namen verwenden, ist es auch einfacher, den AWS Support zu kontaktieren. Sie können sich jedoch dafür entscheiden, sowohl die Richtliniennamen als auch deren Inhalt zu ändern. Weitere Informationen zu IAM-Richtlinien finden Sie unter Erstellen einer vom Kunden verwalteten Richtlinie im IAM-Benutzerhandbuch.

Nachdem Sie die für die Verwendung erforderlichen Richtlinien erstellt haben DataBrew, fügen Sie sie Benutzern und Rollen hinzu. Wie das geht, wird später in diesem Abschnitt beschrieben.

Themen

- Hinzufügen einer IAM-Richtlinie für einen Konsolenbenutzer
- Hinzufügen von Berechtigungen für Datenressourcen für eine IAM-Rolle
- Konfiguration von IAM-Richtlinien für DataBrew

Hinzufügen einer IAM-Richtlinie für einen Konsolenbenutzer

Das Einrichten von Benutzerberechtigungen für den AWS-Managementkonsole ist optional. Wenn Sie jedoch Konsolenzugriff benötigen, führen Sie diesen Schritt zuerst aus.

Um Zugriffsberechtigungen für die Konsole einzurichten, wählen Sie eine der folgenden Optionen: DataBrew

- Verwenden Sie die Richtlinie, die verwaltet wird von AWS:`AwsGlueDataBrewFullAccessPolicy`. Wenn Sie diese Option wählen, fahren Sie mit der nächsten Richtlinie fort,Hinzufügen von Berechtigungen für Datenressourcen für eine IAM-Rolle.
- Erstellen Sie die in diesem Abschnitt beschriebene Richtlinie,`AwsGlueDataBrewCustomUserPolicy`. Mit dieser Option können Sie die Richtlinie mit zusätzlichen benutzerdefinierten Sicherheitsanforderungen anpassen.

Die folgende Richtlinie gewährt die zum Ausführen der DataBrew Konsole erforderlichen Berechtigungen. Sie stellen diese Berechtigungen mithilfe von IAM bereit.

Um die AwsGlueDataBrewCustomUserPolicy IAM-Richtlinie für DataBrew (Konsole) zu definieren

1. Laden Sie die JSON-Datei für die `AwsGlueDataBrewCustomUserPolicy`IAM-Richtlinie herunter.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
3. Wählen Sie im Navigationsbereich Richtlinien.
4. Wählen Sie für jede Richtlinie die Option Create Policy aus.
5. Navigieren Sie auf dem Bildschirm „Richtlinie erstellen“ zur Registerkarte JSON.
6. Kopieren Sie die JSON-Richtlinienanweisung, die Sie heruntergeladen haben. Fügen Sie es über die Beispielanweisung im Editor ein.
7. Vergewissern Sie sich, dass die Richtlinie an Ihr Konto, Ihre Sicherheitsanforderungen und die benötigten AWS Ressourcen angepasst ist. Wenn Sie Änderungen vornehmen müssen, können Sie diese im Editor vornehmen.
8. Wählen Sie Richtlinie prüfen.

Um die AwsGlueDataBrewCustomUserPolicy IAM-Richtlinie zu definieren für DataBrew (AWS CLI)

1. Laden Sie die JSON-Datei für die `AwsGlueDataBrewCustomUserPolicy`IAM-Richtlinie herunter.
2. Passen Sie die Richtlinie wie im ersten Schritt des vorherigen Verfahrens beschrieben an.
3. Führen Sie den folgenden Befehl aus, um die Richtlinie zu erstellen.

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

Hinzufügen von Berechtigungen für Datenressourcen für eine IAM-Rolle

Um eine Verbindung zu Daten herzustellen,AWS Glue DataBrew muss es über eine IAM-Rolle verfügen, die es im Namen des Benutzers weitergeben kann. Im Folgenden erfahren Sie, wie Sie die Richtlinie erstellen, die Sie später einer IAM-Rolle zuordnen.

Die `AwsGlueDataBrewDataResourcePolicy` Richtlinie gewährt die erforderlichen Berechtigungen, um mithilfe DataBrew von Daten eine Verbindung herzustellen. Für jeden Vorgang,

der auf Daten in einer anderen AWS Ressource zugreift, z. B. für den Zugriff auf Ihre Objekte in Amazon S3, ist die Genehmigung DataBrew erforderlich, in Ihrem Namen auf die Ressource zuzugreifen.

Um die AwsGlueDataBrewDataResourcePolicy IAM-Richtlinie für DataBrew (Konsole) zu definieren

1. Laden Sie den JSON-Code für `AwsGlueDataBrewDataResourcePolicy` herunter.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
3. Wählen Sie im Navigationsbereich Richtlinien.
4. Wählen Sie für jede Richtlinie die Option Create Policy aus.
5. Navigieren Sie auf dem Bildschirm „Richtlinie erstellen“ zur Registerkarte JSON.
6. Kopieren Sie die JSON-Richtlinienanweisung, die Sie heruntergeladen haben. Fügen Sie es über die Beispielanweisung im Editor ein.
7. Vergewissern Sie sich, dass die Richtlinie an Ihr Konto, Ihre Sicherheitsanforderungen und die benötigten AWS Ressourcen angepasst ist. Wenn Sie Änderungen vornehmen müssen, können Sie diese im Editor vornehmen.
8. Wählen Sie Richtlinie prüfen.

Um die AwsGlueDataBrewDataResourcePolicy IAM-Richtlinie zu definieren für DataBrew (AWS CLI)

1. Laden Sie die JSON-Datei für `AwsGlueDataBrewDataResourcePolicy` herunter.
2. Passen Sie die Richtlinie wie im ersten Schritt des vorherigen Verfahrens beschrieben an.
3. Führen Sie den folgenden Befehl aus, um die Richtlinie zu erstellen.

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

Konfiguration von IAM-Richtlinien für DataBrew

Im Folgenden finden Sie Details und Beispiele zu IAM-Richtlinien, die Sie zusammen verwenden können. DataBrew Einzelheiten zu den grundlegenden Richtlinien finden Sie hier. Außerdem gibt es weitere Beispiele, deren Verwendung nicht erforderlich ist DataBrew. Dies sind zusätzliche Konfigurationen, die Sie in bestimmten Situationen verwenden könnten.

Themen

- AwsGlueDataBrewCustomUserPolicy
- AwsGlueDataBrewDataResourcePolicy
- IAM-Richtlinie zur Verwendung von Amazon S3 S3-Objekten mit DataBrew
- IAM-Richtlinie, mit der Verschlüsselung verwendet werden soll DataBrew

AwsGlueDataBrewCustomUserPolicy

Die `AwsGlueDataBrewCustomUserPolicy` Richtlinie gewährt die meisten Berechtigungen, die für die Verwendung der DataBrew Konsole erforderlich sind. Einige der in dieser Richtlinie angegebenen Ressourcen beziehen sich auf Dienste, die von verwendet werden DataBrew. Dazu gehören Namen für AWS Glue Data Catalog Amazon S3 S3-Buckets, Amazon CloudWatch Logs und AWS KMS Ressourcen. Sie ähnelt der genannten AWS-verwalteten Richtlinie. `AwsGlueDataBrewFullAccessPolicy`

Die folgende Tabelle beschreibt die Berechtigungen, die von dieser Richtlinie erteilt werden.

Action (Aktion)	Ressource	Beschreibung
`"databrew:*"`	`"*"`	Erteilt die Erlaubnis, alle DataBrew API-Operationen auszuführen.

`"glue:GetDatabases"`
`"glue:GetPartitions"`
`"glue:GetTable"`
`"glue:GetTables"`
`"glue:GetDataCatalogEncryptionSettings"`
	`"*"`	Ermöglicht das Auflisten von AWS Glue Datenbanken und Tabellen.

`"dataexchange:ListDataSets"` `"dataexchange:ListDataSetRevisions"`	`"*"`	Ermöglicht das Auflisten von AWS Datenaustauschressourcen in Datensätzen.

Action (Aktion)	Ressource	Beschreibung
"dataexchange:List RevisionAssets" "dataexchange:Crea teJob" "dataexchange:StartJob" "dataexchange:GetJob"		
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	Ermöglicht die Auflistung von AWS KMS Schlüsseln, die für die Verschlüsselung der Jobausgabe verwendet werden sollen.
"kms:GenerateDataKey"	"arn:aws:kms:::key/ key_ids"	Ermöglicht die Verschlüs selung der Jobausgabe.
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionC onfiguration"	"arn:aws:s3:::buck et_name/*", "arn:aws:s3:::buck et_name"	Ermöglicht das Auflisten von Amazon S3 S3-Bucket s für Projekte, Datensätze und Jobs. Ermöglicht das Senden von Ausgabeda teien an S3.
"sts:GetCallerIdentity"	"*"	Ruft Informationen über den aktuellen Anrufer ab.
"cloudtrail:Lookup Events",	"*"	Erlaubt das Auflisten von AWS CloudTrail Ereigniss en für Datensätze (Data Lineage).

Action (Aktion)	Ressource	Beschreibung
`"iam:ListRoles"` `"iam:GetRole"`	`"*"`	Ermöglicht das Auflisten von IAM-Rollen, die für Projekte und Jobs verwendet werden können.

AwsGlueDataBrewDataResourcePolicy

Die `AwsGlueDataBrewDataResourcePolicy` Richtlinie gewährt die Berechtigungen, die zum Herstellen einer Verbindung mit Daten und zur Konfiguration DataBrew erforderlich sind.

Die folgende Tabelle beschreibt die Berechtigungen, die von dieser Richtlinie erteilt werden.

Action (Aktion)	Ressource	Beschreibung
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Ermöglicht es Ihnen, eine Vorschau Ihrer Dateien anzuzeigen.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Ermöglicht das Senden von Ausgabedateien an S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Ermöglicht das Löschen eines Objekts, das von erstellt wurde DataBrew.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*",` `"arn:aws:s3:::bucket_name"`	Ermöglicht das Auflisten von Amazon S3 S3-Buckets aus Projekten, Datensätzen und Jobs.

Action (Aktion)	Ressource	Beschreibung
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Ermöglicht das Entschlüsseln verschlüsselter Datensätze.
`"kms:GenerateDataKey"`	`"arn:aws:kms:::key/key_ids"`	Ermöglicht die Verschlüsselung der Jobausgabe.

`"ec2:DescribeVpcEndpoints"`
`"ec2:DescribeRouteTables"`
`"ec2:DeleteNetworkInterface"`
`"ec2:DescribeNetworkInterfaces"`
`"ec2:DescribeSecurityGroups"`
`"ec2:DescribeSubnets"`
`"ec2:DescribeVpcAttribute"`
`"ec2:CreateNetworkInterface"`
	`"*"`	Ermöglicht die Einrichtung von Amazon EC2 EC2-Netzwerkelementen wie Virtual Private Clouds (VPCs) bei der Ausführung von Jobs und Projekten.

`"ec2:DeleteNetworkInterface"`	`"*"`	Ermöglicht das Löschen einer Netzwerkschnittstelle in einer VPC.

Action (Aktion)	Ressource	Beschreibung

`"ec2:CreateTags"`
 	
`"ec2:DeleteTags"`
	`"arn:aws:ec2:::network-interface/*", "arn:aws:ec2:::security-group/*"`	Ermöglicht das Erstellen und Löschen von Tags.
		
		Sie benötigen diese Berechtigungen, wenn Sie einen AWS Glue Datenkatalog mit aktivierter VPC verwenden. DataBrew leitet Daten weiter,AWS Glue um Ihre Jobs und Projekte auszuführen. Diese Berechtigungen ermöglichen das Markieren von Amazon EC2 EC2-Ressourcen, die für Entwicklungsendpunkte erstellt wurden.AWS Glue kennzeichnet Amazon EC2 EC2-Netzwerkschnittstellen, Sicherheitsgruppen und Instances mit`aws-glue-service-resource` .

`"logs:CreateLogGroup"`
 	
`"logs:CreateLogStream"`
 	
`"logs:PutLogEvents"`
	`"arn:aws:logs:::log-group:/aws-glue-databrew/*"`	Ermöglicht das Schreiben von Protokollen in Amazon CloudWatch Logs
		
		DataBrew schreibt Protokolle in Protokollgruppen, deren Namen mit beginnen`aws-glue-databrew` .

Action (Aktion)	Ressource	Beschreibung
`"lakeformation:GetDataAccess"`	`"*"`	Erlaubt den Zugriff auf AWS Lake Formation, sofern er auch erlaubt `"Glue":"GetTable"` ist Die Verwendung von Lake Formation erfordert eine weitere Konfiguration in der Lake Formation Formation-Konsole.

IAM-Richtlinie zur Verwendung von Amazon S3 S3-Objekten mit DataBrew

Die `AwsGlueDataBrewSpecificS3BucketPolicy` Richtlinie gewährt Benutzern ohne Administratorrechte die für den Zugriff auf S3 erforderlichen Berechtigungen.

Passen Sie die Richtlinie wie folgt an:

1. Ersetzen Sie die Amazon S3 S3-Pfade in der Richtlinie so, dass sie auf die Pfade verweisen, die Sie verwenden möchten. Steht im Beispieltext *BUCKET-NAME-1/SPECIFIC-OBJECT-NAME* für ein bestimmtes Objekt oder eine bestimmte Datei. *BUCKET-NAME-2/*steht für alle Objekte (*), deren Pfadname mit beginntBUCKET-NAME-2/. Aktualisieren Sie diese, um die Buckets zu benennen, die Sie verwenden.
2. (Optional) Verwenden Sie Platzhalter in den Amazon S3 S3-Pfaden, um die Berechtigungen weiter einzuschränken. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags (Markierungen) im IAM-Benutzerhandbuch.

   Bewährte Sicherheitsmethode: Um unbefugten Zugriff auf Amazon S3 S3-Buckets mit ähnlichen Namen in anderen AWS Konten zu verhindern, nehmen Sie den `aws:ResourceAccount` Bedingungsschlüssel in Ihre Richtlinie auf. Dadurch wird sichergestellt, dass DataBrew Sie nur auf Buckets innerhalb Ihres eigenen AWS Kontos zugreifen können, selbst wenn Sie Wildcard-Ressourcen-ARNs verwenden. Fügen Sie Ihren Richtlinienerklärungen die folgende Bedingung hinzu:

```
"Condition": {
"StringEquals": {
```

```
"aws:ResourceAccount": "123456789012"
}
}
```

`123456789012`Ersetzen Sie es durch Ihre tatsächliche AWS Konto-ID.

In diesem Zusammenhang können Sie die Berechtigungen für die Aktionen `s3:PutObject` und einschränken`s3:PutBucketCORS`. Diese Aktionen sind nur für Benutzer erforderlich, die DataBrew Projekte erstellen, da diese Benutzer in der Lage sein müssen, Ausgabedateien an S3 zu senden.

Weitere Informationen und einige Beispiele dafür, was Sie zu einer IAM-Richtlinie für Amazon S3 hinzufügen können, finden Sie unter Beispiele für Bucket-Richtlinien im Amazon S3 S3-Entwicklerhandbuch.

Die folgende Tabelle beschreibt die Berechtigungen, die von dieser Richtlinie erteilt werden.

Action (Aktion)	Ressource	Beschreibung
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht es Ihnen, eine Vorschau Ihrer Dateien anzuzeigen.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht das Senden von Ausgabedateien an S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht das Löschen eines Objekts.

Um die AwsGlueDataBrewSpecificS3BucketPolicy IAM-Richtlinie für DataBrew (Konsole) zu definieren

1. Laden Sie die JSON-Datei für die `AwsGlueDataBrewSpecificS3BucketPolicy`IAM-Richtlinie herunter.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
3. Wählen Sie im Navigationsbereich Richtlinien.
4. Wählen Sie für jede Richtlinie die Option Create Policy aus.
5. Navigieren Sie auf dem Bildschirm „Richtlinie erstellen“ zur Registerkarte JSON.
6. Fügen Sie die JSON-Anweisung für die Richtlinie über die Beispielanweisung im Editor ein.
7. Stellen Sie sicher, dass die Richtlinie an Ihr Konto, Ihre Sicherheitsanforderungen und die erforderlichen AWS Ressourcen angepasst ist. Wenn Sie Änderungen vornehmen müssen, können Sie diese im Editor vornehmen.
8. Wählen Sie Richtlinie prüfen.

Um die AwsGlueDataBrewSpecificS3BucketPolicy IAM-Richtlinie zu definieren für DataBrew (AWS CLI)

1. Laden Sie die JSON-Datei für `AwsGlueDataBrewSpecificS3BucketPolicy`herunter.
2. Passen Sie die Richtlinie wie im ersten Schritt des vorherigen Verfahrens beschrieben an.
3. Führen Sie den folgenden Befehl aus, um die Richtlinie zu erstellen.

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

IAM-Richtlinie, mit der Verschlüsselung verwendet werden soll DataBrew

Die `AwsGlueDataBrewS3EncryptedPolicy` Richtlinie gewährt Benutzern ohne Administratorrechte die Berechtigungen, die für den Zugriff auf mit AWS Key Management Service(AWS KMS) verschlüsselte S3-Objekte erforderlich sind.

Passen Sie die Richtlinie wie folgt an:

1. Ersetzen Sie die Amazon S3 S3-Pfade in der Richtlinie so, dass sie auf die Pfade verweisen, die Sie verwenden möchten. Steht im Beispieltext *BUCKET-NAME-1/SPECIFIC-OBJECT-NAME* für ein bestimmtes Objekt oder eine bestimmte Datei. *BUCKET-NAME-2/*steht für alle Objekte (*), deren Pfadname mit beginntBUCKET-NAME-2/. Aktualisieren Sie diese, um die Buckets zu benennen, die Sie verwenden.
2. (Optional) Verwenden Sie Platzhalter in den Amazon S3 S3-Pfaden, um die Berechtigungen weiter einzuschränken. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags.

   In diesem Zusammenhang können Sie die Berechtigungen für die Aktionen `s3:PutObject` und `s3:PutBucketCORS` einschränken. Diese Aktionen sind nur für Benutzer erforderlich, die DataBrew Projekte erstellen, da diese Benutzer in der Lage sein müssen, Ausgabedateien an S3 zu senden.

   Weitere Informationen und einige Beispiele dafür, was Sie zu einer IAM-Richtlinie für Amazon S3 hinzufügen können, finden Sie unter Beispiele für Bucket-Richtlinien.
3. Suchen Sie in der Datei nach den folgenden Ressourcen-ARNs. `ToUseKms`

   ```
     "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. Ändern Sie das AWS Beispielkonto in Ihre AWS Kontonummer (ohne Bindestriche).
5. Ändern Sie die Beispielliste so, dass sie stattdessen die IAM-Rollen auflistet, die Sie verwenden möchten. Wir empfehlen, Ihre IAM-Richtlinien auf den kleinstmöglichen Berechtigungssatz zu beschränken. Sie können Ihrem Benutzer jedoch Zugriff auf alle IAM-Rollen gewähren, wenn Sie beispielsweise ein persönliches Lernkonto mit Beispieldaten verwenden. Damit die Liste auf alle IAM-Rollen zugreifen kann, ändern Sie die Beispielliste in einen Eintrag:. `"arn:aws:iam::111122223333:role/*"`

Die folgende Tabelle beschreibt die Berechtigungen, die von dieser Richtlinie erteilt werden.

Action (Aktion)	Ressource	Beschreibung
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht es Ihnen, eine Vorschau Ihrer Dateien anzuzeigen.

Action (Aktion)	Ressource	Beschreibung
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht das Auflisten von Amazon S3 S3-Buckets aus Projekten, Datensätzen und Jobs.
`"s3:PutObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht das Senden von Ausgabedateien an S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Ermöglicht das Löschen eines Objekts, das von erstellt wurde DataBrew.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Ermöglicht das Entschlüsseln verschlüsselter Datensätze.
`"kms:GenerateDataKey*"`	`"arn:aws:kms:::key/key_ids"`	Ermöglicht die Verschlüsselung der Jobausgabe.

Um die AwsGlueDataBrewS3EncryptedPolicy IAM-Richtlinie für DataBrew (Konsole) zu definieren

1. Laden Sie die JSON-Datei für die `AwsGlueDataBrewS3EncryptedPolicy`IAM-Richtlinie herunter.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
3. Wählen Sie im Navigationsbereich Richtlinien.
4. Wählen Sie für jede Richtlinie die Option Create Policy aus.
5. Navigieren Sie auf dem Bildschirm „Richtlinie erstellen“ zur Registerkarte JSON.
6. Fügen Sie die JSON-Anweisung für die Richtlinie über die Beispielanweisung im Editor ein.

7. Stellen Sie sicher, dass die Richtlinie an Ihr Konto, Ihre Sicherheitsanforderungen und die erforderlichen AWS Ressourcen angepasst ist. Wenn Sie Änderungen vornehmen müssen, können Sie diese im Editor vornehmen.
8. Wählen Sie Richtlinie prüfen.

Um die AwsGlueDataBrewS3EncryptedPolicy IAM-Richtlinie zu definieren für DataBrew (AWS CLI)

1. Laden Sie die JSON-Datei für `AwsGlueDataBrewS3EncryptedPolicy`herunter.
2. Passen Sie die Richtlinie wie im ersten Schritt des vorherigen Verfahrens beschrieben an.
3. Führen Sie den folgenden Befehl aus, um die Richtlinie zu erstellen.

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

Benutzer oder Gruppen mit DataBrew Berechtigungen hinzufügen

Sie weisen Rollen Richtlinien und Benutzern und Gruppen Rollen zu, um Berechtigungen zu verwalten. Weitere Informationen finden Sie unter IAM-Identitäten (Benutzer, Gruppen und Rollen) im IAM-Benutzerhandbuch.

Bevor Sie beginnen, benötigen Sie mindestens einen Benutzer, dem Sie Berechtigungen zuweisen können.

Gehen Sie wie folgt vor, um DataBrew Berechtigungen für Benutzer einzurichten, die in der DataBrew Konsole arbeiten oder DataBrew Befehle in der CLI ausführen müssen.

So richten Sie DataBrew Berechtigungen ein

1. Erstellen Sie einen Zugriffsschlüssel, mit dem Ihr Benutzer das AWS CLI for DataBrew und andere Entwicklungstools verwenden kann.
2. Aktivieren AWS-Managementkonsole Sie den Zugriff, damit der Benutzer die AWS Konsole verwenden kann.
3. Erstellen Sie eine Rolle für DataBrew Benutzer oder Gruppen.
4. Wählen Sie die Richtlinie aus, die Sie verwenden. Führen Sie eine der folgenden Aktionen aus:

- Wenn Sie sie erstellt habenAwsGlueDataBrewCustomUserPolicy, wählen Sie sie aus der Liste aus.
- Um die AWS-managed Richtlinie zu verwenden, wählen Sie sie AwsGlueDataBrewFullAccessPolicy aus der Liste aus.

5. Weisen Sie diese Richtlinie der Rolle zu.
6. Stellen Sie die Vertrauensstellungen für die Rolle so ein, dass ein Benutzer oder eine Gruppe die entsprechende Rolle übernehmen kann.

   - Wenn Sie keine Gruppen verwenden, vertrauen Sie dem Benutzer die Rolle an.
   - Wenn Sie Gruppen verwenden, vertrauen Sie der Gruppe die Rolle an und fügen Sie den Benutzer der Gruppe hinzu.

Hinzufügen einer IAM-Rolle mit Datenressourcenberechtigungen

Sie verwenden IAM-Rollen, um gemeinsam zugewiesene Richtlinien zu verwalten. Eine IAM-Rolle kann von jemandem verwendet werden, der eine bestimmte Rolle innehat, z. B. von einem DataBrew Benutzer oder DataBrew von sich selbst. Weitere Informationen finden Sie unter IAM-Rollen im IAM-Benutzerhandbuch.

Gehen Sie wie folgt vor, um eine IAM-Rolle zu erstellen, die für den Zugriff von DataBrew Projekten auf Daten erforderlich ist.

So fügen Sie die erforderliche IAM-Richtlinie einer neuen IAM-Rolle hinzu für DataBrew

1. Wählen Sie im Navigationsbereich Roles (Rollen) und Create Role (Rolle erstellen) aus.
2. Wählen Sie unter Typ der vertrauenswürdigen Entität auswählen die Karte mit der Bezeichnung AWS Service aus.
3. Wählen Sie DataBrewaus der Liste aus und klicken Sie dann auf Weiter: Berechtigungen.
4. Geben Sie **`AwsGlueDataBrewDataResourcePolicy`** in das Suchfeld ein (die IAM-Richtlinie, die Sie in einem früheren Schritt erstellt haben). Wählen Sie die Richtlinie aus und klicken Sie auf Weiter: Tags.
5. Wählen Sie Weiter: Prüfen aus.
6. Geben Sie für Rollenname den Namen **`AwsGlueDataBrewDataAccessRole`** ein und klicken Sie auf Rolle erstellen.

Einrichtung AWS IAM Identity Center(IAM Identity Center)

Mithilfe von AWS IAM Identity Center(IAM Identity Center) können sich Ihre Benutzer DataBrew mit einer einfachen URL anmelden, ohne sich bei der anzumelden AWS-Managementkonsole und ohne ein Konto zu benötigen.AWS

So richten Sie IAM Identity Center ein

1. Öffnen Sie die AWS Organizations Konsole und erstellen Sie eine Organisation, falls Sie noch keine haben. Alle Funktionen sind standardmäßig für diese Organisation aktiviert.

   Weitere Informationen finden Sie unter AWS IAM Identity Center Voraussetzungen und Organisation erstellen und verwalten.
2. Öffnen Sie die AWS IAM Identity Center-Konsole.
3. Wählen Sie Ihre Identitätsquelle.

   Standardmäßig erhalten Sie einen IAM Identity Center Store für eine schnelle und einfache Benutzerverwaltung. Optional können Sie stattdessen eine Verbindung zu einem externen Identitätsanbieter herstellen oder ein AWS Managed Microsoft AD Verzeichnis mit Ihrem lokalen Active Directory verbinden. In diesem Handbuch verwenden wir den standardmäßigen IAM Identity Center-Speicher.

   Weitere Informationen finden Sie unter Wählen Sie Ihre Identitätsquelle im AWS IAM Identity Center Benutzerhandbuch.
4. Erstellen Sie einen Berechtigungssatz für DataBrew den Zugriff:

   a. Wählen Sie im Navigationsbereich von IAM Identity Center AWS Konten und dann Berechtigungssätze aus.
   b. Wählen Sie auf der Seite Berechtigungssatz erstellen die Option Benutzerdefinierten Berechtigungssatz erstellen aus.
   c. Geben Sie für Relay-Status den Wert ein`https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`.

      Wenn Sie diesen Wert eingeben, können Ihre Benutzer direkt zu DataBrew.
   d. Wählen Sie AWS Verwaltete Richtlinien anhängen DataBrew, suchen Sie nach und wählen Sie AwsGlueDataBrewFullAccessPolicy. Wenn Sie diese Option wählen, erhalten Ihre Benutzer alle Berechtigungen, die sie benötigen DataBrew. Weitere Informationen finden Sie unterHinzufügen einer IAM-Richtlinie für einen Konsolenbenutzer.

e. (Optional) Wählen Sie Benutzerdefinierte Berechtigungsrichtlinie erstellen und passen Sie die Berechtigungen für Ihre Benutzer an.

5. Wählen Sie im Navigationsbereich von IAM Identity Center Gruppen und anschließend Gruppe erstellen aus. Geben Sie den Gruppennamen ein und wählen Sie Create aus.
6. Fügen Sie einen Benutzer zum IAM Identity Center Store hinzu:

   a. Wählen Sie im Navigationsbereich von IAM Identity Center die Option Benutzer aus.

   b. Geben Sie auf dem Bildschirm „Benutzer hinzufügen“ die erforderlichen Informationen ein und wählen Sie „Dem Benutzer eine E-Mail mit Anweisungen zur Einrichtung des Passworts senden“. Der Benutzer sollte eine E-Mail mit den nächsten Einrichtungsschritten erhalten.

   c. Wählen Sie Weiter: Gruppen, wählen Sie die gewünschte Gruppe aus und klicken Sie auf Benutzer hinzufügen.

      Benutzer sollten eine E-Mail erhalten, in der sie zur Verwendung von SSO eingeladen werden. In dieser E-Mail müssen sie Einladung annehmen auswählen und das Passwort festlegen. Sie können die Portal-URL auch in der E-Mail finden. Sie können diese URL für den Zugriff verwenden DataBrew.

7. Ordnen Sie jeden Benutzer einem Konto zu:

   a. Öffnen Sie die IAM Identity Center-Konsole und wählen Sie im Navigationsbereich AWS Konten aus.

   b. Wählen Sie AWS Organisation und wählen Sie ein AWS Konto aus.

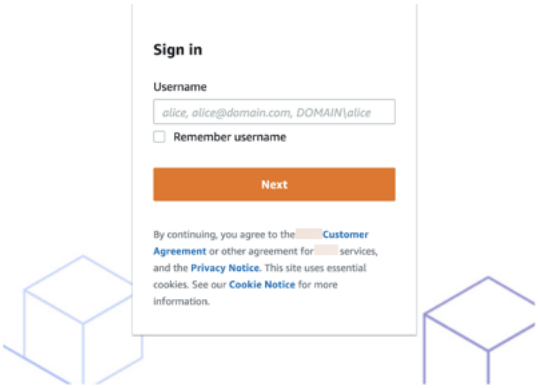
   c. Wählen Sie auf dem Bildschirm „Benutzer zuweisen“ die Registerkarte „Gruppen“ und wählen Sie die gewünschte Gruppe aus.

   d. Wählen Sie Next: Permission sets (Weiter: Berechtigungssätze) aus.

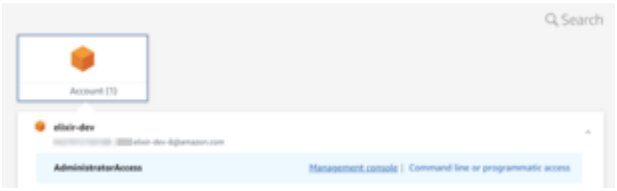
   e. Wählen Sie den Berechtigungssatz für und DataBrew klicken Sie auf Fertig stellen.

Anmeldeschritte für einen IAM Identity-Benutzer Center-enabled

1. Melden Sie sich AWS mit einem IAM Center-enabled Identity-Konto an.



2. Klicken Sie auf AWS Kontoidentität



3. Klicken Sie auf Managementkonsole, um mit einem Klick zur Konsole umzuleiten. DataBrew

Verwendung DataBrew als Erweiterung in JupyterLab

> **Warning**
>
> AWS Glue DataBrew JupyterLab Der Support für Erweiterungen endet am 31. Dezember 2024, da der Support für JupyterLab 3 Personen endet. Weitere Informationen finden Sie unter JupyterLab 3 Ende der Wartung.

Wenn Sie es vorziehen, Daten in einer Jupyter Notebook-Umgebung vorzubereiten, können Sie alle Funktionen von in nutzen.AWS Glue DataBrew JupyterLab

JupyterLab ist eine webbasierte interaktive Entwicklungsumgebung für Jupyter Notebook. Auf der lokalen JupyterLab Webseite können Sie Abschnitte für ein Terminal, eine SQL-Sitzung, Python und mehr hinzufügen. Nach der Installation der AWS Glue DataBrew Erweiterung können Sie einen Abschnitt für die DataBrew Konsole hinzufügen. Es läuft mit allen vorhandenen Notebooks oder anderen Erweiterungen, die Sie bereits haben, direkt aus der JupyterLab Umgebung.

Themen

- Voraussetzungen

- Konfiguration JupyterLab für die Verwendung der Erweiterung
- Aktivierung der Erweiterung für DataBrew JupyterLab

Voraussetzungen

Bevor Sie beginnen, richten Sie die folgenden Elemente ein:

- Ein AWS Konto — Wenn Sie noch keins haben, beginnen Sie mitEin neues einrichten AWS Konto.
- Ein AWS Identity and Access Management(IAM-) Benutzer mit Zugriff auf die erforderlichen Berechtigungen für DataBrew — Weitere Informationen finden Sie unterBenutzer oder Gruppen mit DataBrew Berechtigungen hinzufügen.
- Eine IAM-Rolle zur Verwendung im DataBrew Betrieb — Sie können die Standardrolle verwenden, sofern sie konfiguriert `AwsGlueDataBrewDataAccessRole` ist. Informationen zum Einrichten zusätzlicher IAM-Rollen finden Sie unter. Hinzufügen einer IAM-Rolle mit Datenressourcenberechtigungen
- Eine JupyterLab Installation (Version 2.2.6 oder höher) — Weitere Informationen finden Sie in der Dokumentation zu den folgenden Themen: JupyterLab
  - JupyterLab Voraussetzungen
  - JupyterLab Installation — Wir empfehlen die Verwendung von`pip install jupyterlab`.
- Eine Node.js Installation (Version 12.0 oder höher).
- Eine AWS Command Line Interface(AWS CLI) Installation — Weitere Informationen finden Sie unterEinrichtung der AWS CLI.
- Eine AWS Jupyter-Proxyinstallation (`pip install aws-jupyter-proxy`) — Diese Erweiterung wird mit einem AWS Service-Endpunkt verwendet, um Ihre Anmeldeinformationen sicher weiterzuleiten.AWSWeitere Informationen finden Sie unter aws-jupyter-proxy on. GitHub

Um zu überprüfen, ob Sie die erforderlichen Komponenten installiert haben, können Sie in der Befehlszeile einen Test ausführen, der dem folgenden ähnelt, wie im folgenden Beispiel gezeigt.

```
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
```

Die Ausgabe sollte etwa wie folgt aussehen. Die Verzeichnisse variieren je nach Betriebssystem und Konfiguration.

```
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
```

Konfiguration JupyterLab für die Verwendung der Erweiterung

Nach der Installation müssen Sie sie konfigurieren JupyterLab, um den Datenzugriff zu sichern und Servererweiterungen zu aktivieren.

Um ein Passwort und eine Verschlüsselung zu konfigurieren

1. Legen Sie ein Passwort fest, um die Daten zu schützen, die Sie der Erweiterung hinzufügen möchten. Jupyter bietet ein Passwort-Dienstprogramm. Führen Sie den folgenden Befehl aus und geben Sie Ihr bevorzugtes Passwort ein, wenn Sie dazu aufgefordert werden.

   ```
   jupyter notebook password
   ```

Die Ausgabe sieht ungefähr wie folgt aus.

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

2. Aktivieren Sie die Verschlüsselung auf dem Jupyter-Server. Wenn Sie Jupyter auf Ihrem lokalen Computer installieren und niemand über das Netzwerk darauf zugreifen kann, können Sie diesen Schritt überspringen.

   Um die Verschlüsselung mit Transport Layer Security (TLS) einzurichten, erstellen Sie ein auf Ihre Umgebung zugeschnittenes Zertifikat. Weitere Informationen finden Sie unter Verwenden von Let's Encrypt bei der Sicherung eines Servers in der Jupyter-Dokumentation.

3. Führen Sie zunächst JupyterLab den folgenden Befehl an der Befehlszeile aus.

   ```
   jupyter lab
   ```

   Weitere Informationen finden Sie JupyterLab in der JupyterLab Dokumentation unter Starten.

4. Während JupyterLab der Ausführung können Sie über eine URL darauf zugreifen, die der folgenden ähnelt: http://localhost:8888/lab. Wenn Sie die Verschlüsselung einrichten, verwenden Sie `https` anstelle von`http`. Wenn Sie den Port angepasst haben, ersetzen Sie stattdessen Ihre Portnummer`8888`.

Gehen Sie wie folgt vor, um die Erweiterungen von Drittanbietern zu aktivieren.

So aktivieren Sie Erweiterungen von Drittanbietern in JupyterLab

1. Wählen Sie auf der JupyterLab Webseite im Menü links das Extension Manager-Symbol aus.
2. Lesen Sie die Warnung zu den Risiken, die mit der Ausführung von Erweiterungen von Drittanbietern verbunden sind. Installieren Sie nur Erweiterungen von Entwicklern, denen Sie vertrauen.
3. Um Erweiterungen von Drittanbietern zu aktivieren JupyterLab, wählen Sie Aktivieren.
4. Folgen Sie den Anweisungen zur Wiederherstellung und zum erneuten JupyterLab Laden.

Aktivierung der Erweiterung für DataBrew JupyterLab

Nachdem Sie eine sichere Installation von JupyterLab mit aktivierten Erweiterungen durchgeführt haben, installieren Sie die DataBrew Erweiterung, damit Sie sie DataBrew in Ihrem Notizbuch ausführen können.

Um die Erweiterungen für DataBrew (Konsole) zu installieren

1. Führen Sie zunächst JupyterLab den folgenden Befehl an der Eingabeaufforderung aus.

   ```
   jupyter lab
   ```

2. Wählen Sie auf der JupyterLab Webseite im Menü links das Extension Manager-Symbol aus.
3. Suchen Sie nach der DataBrew Erweiterung, indem Sie oben links **`brew`** „" für Suchen eingeben.
4. Suchen Sie aws_glue_databrew_jupyter in der Liste, aber klicken Sie nicht darauf. Wenn Sie auf den hervorgehobenen Namen der Erweiterung klicken, wird ein neues Browserfenster mit der Seite aws_glue_databrew_jupyter geöffnet. GitHub
5. Um die Erweiterung zu installieren, wählen Sie eine der folgenden Optionen: DataBrew

   - Führen Sie in der Befehlszeile den Befehl aus`jupyter labextension install aws_glue_databrew_jupyter`.
   - Wählen Sie auf der Unterseite der Erweiterungskarte unter "aws_glue_databrew_jupyter" in grauer Schrift die Option Installieren.

   DataBrew Die JupyterLab Erweiterung ist mit den Versionen 1.2 und 2.x kompatibel.
6. Führen `jupyter labextension list` Sie den Befehl aus, um zu überprüfen, ob es installiert ist. Die Ausgabe sollte etwa wie folgt aussehen.

   ```
   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. Verwenden Sie eine der folgenden Methoden zur JupyterLab Neuerstellung:

   - Führen Sie in der Befehlszeile den Befehl aus`jupyter lab build`.
   - Wählen Sie auf der Webseite oben links die Option Neu erstellen aus.
8. Wenn der Build abgeschlossen ist, führen Sie einen der folgenden Schritte aus:

- Führen Sie in der Befehlszeile den Befehl aus`jupyter lab`.
- Wählen Sie auf der Webseite in der Meldung Build Complete die Option Reload aus.

9. Schließen Sie auf der JupyterLab Webseite den Extension Manager, indem Sie das entsprechende Symbol im Menü auf der linken Seite auswählen.

   Um die Erweiterung zu öffnen, wählen Sie auf der Registerkarte Launcher im Bereich Andere die Option Starten AWS Glue DataBrew aus. Die Erweiterung verwendet Ihre aktuelle AWS CLI Konfiguration für Zugriffstasten und AWS Regionseinstellungen.

Nachdem Sie die Einrichtung abgeschlossen haben, können Sie die AWS Glue DataBrewRegisterkarte verwenden, um DataBrew von innen heraus mit ihnen zu interagieren JupyterLab.

Erste Schritte mit AWS Glue DataBrew

Sie können das folgende Tutorial verwenden, um sich bei der Erstellung Ihres ersten DataBrew Projekts zu unterstützen. Sie laden einen Beispieldatensatz, führen Transformationen für diesen Datensatz aus, erstellen ein Rezept für die Erfassung dieser Transformationen und führen einen Job aus, um die transformierten Daten in Amazon S3 zu schreiben.

Themen

- Voraussetzungen
- Schritt 1: Erstellen eines Projekts
- Schritt 2: Fassen Sie die Daten zusammen
- Schritt 3: Weitere Transformationen hinzufügen
- Schritt 4: Überprüfe deine DataBrew Ressourcen
- Schritt 5: Erstellen Sie ein Datenprofil
- Schritt 6: Transformieren Sie den Datensatz
- Schritt 7: (Optional) Aufräumen

Voraussetzungen

Bevor Sie fortfahren, folgen Sie den entsprechenden Anweisungen unter. Einrichtung AWS Glue DataBrew Fahren Sie dann fort mitSchritt 1: Erstellen eines Projekts.

Schritt 1: Erstellen eines Projekts

In diesem Schritt verwenden Sie die DataBrew Konsole, um schnell mit einem Beispielprojekt zu beginnen.

So erstellen Sie ein Projekt

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole unter https://console.aws.amazon.com/databrew/.
2. Vergewissern Sie sich, dass Ihre AWS Region oben rechts auf der DataBrew Konsole ausgewählt ist. Eine Liste der AWS Regionen, die von unterstützt werden DataBrew, finden Sie unter DataBrew Endpunkte und Kontingente in der.Allgemeine AWS-Referenz
3. Wählen Sie im Navigationsbereich Projekte und dann Projekt erstellen aus.

4. Gehen Sie im Bereich Projektdetails wie folgt vor:

   - Geben Sie als Projektname ein`chess-project`.
   - Erstellen Sie für Angehängtes Rezept ein neues Rezept. Es wird ein Namensvorschlag für das Rezept bereitgestellt (`chess-project-recipe`).
5. Wählen Sie im Bereich Datensatz auswählen die Option Beispieldateien aus.
6. Wählen Sie im Bereich Beispieldateien die Option Berühmte Schachzüge aus. Dieser Datensatz enthält detaillierte Informationen zu mehr als 20.000 Schachpartien.

   Für den Datensatznamen wird ein Namensvorschlag für den Datensatz bereitgestellt (`chess-games`).
7. Wählen Sie im Bereich Zugriffsberechtigungen die Option`AwsGlueDataBrewDataAccessRole`. Dies ist eine servicebezogene Rolle, mit der Sie in Ihrem DataBrew Namen auf Ihre Amazon S3 S3-Buckets zugreifen können.
8. Wählen Sie Projekt erstellen und warten Sie, bis die Vorbereitung des Projekts DataBrew abgeschlossen ist. Das Fenster sieht wie folgt aus.

   Die Daten, die Sie sehen, stellen ein Beispiel aus dem `chess-games` Datensatz dar. Standardmäßig besteht die Stichprobe aus den ersten 500 Zeilen des Datensatzes. Sie können diese Projekteinstellung später ändern.

   Die Werkzeugleiste bietet Zugriff auf Hunderte von Datentransformationen, die Sie auf die Daten anwenden können.

   Im Rezeptbereich rechts in der DataBrew Konsole werden die Transformationen nachverfolgt, die Sie bisher angewendet haben.

Schritt 2: Fassen Sie die Daten zusammen

In diesem Schritt erstellen Sie ein DataBrew Rezept — eine Reihe von Transformationen, die auf diesen Datensatz und ähnliche Datensätze angewendet werden können. Wenn das Rezept fertig ist, veröffentlichen Sie es, damit es verwendet werden kann.

Im Schachspiel können Spieler danach bewertet werden, wie gut sie im Vergleich zu anderen Spielern abschneiden. (Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/Chess_rating_system.) In diesem Tutorial konzentrieren Sie sich nur auf die Spiele, bei denen beide Spieler der Klasse A angehörten, was bedeutet, dass ihre Bewertungen 1800 oder mehr betrugen.

Um die Daten zusammenzufassen

1. Wählen Sie auf der Transformationssymbolleiste Filter, Nach Bedingung, Größer als oder gleich aus.
2. Stellen Sie diese Optionen wie folgt ein:

   - Quellspalte - `white_rating`
   - Filterbedingung — Größer als oder gleich 1800

   Um zu sehen, wie die Transformation funktioniert, wählen Sie „Änderungen in der Vorschau anzeigen“. Wählen Sie dann Apply (Anwenden).
3. Wiederholen Sie den vorherigen Schritt, setzen Sie diesmal jedoch die Quellspalte auf`black_rating`. Nachdem Sie Ihre Änderungen übernommen haben, enthalten die Beispieldaten nur die Spiele, in denen die Spieler auf jeder Seite (schwarz und weiß) der Klasse A oder höher angehörten.
4. Fassen Sie die Daten zusammen, um festzustellen, wie viele Spiele von jeder Seite gewonnen wurden. Wählen Sie dazu in der Transformationswerkzeugleiste die Option Gruppe aus.
5. Gehen Sie für die Gruppeneigenschaften wie folgt vor:

   a. Wählen Sie in der ersten Zeile `winner` den Namen der Spalte aus. Lassen Sie Aggregieren auf Gruppieren nach eingestellt.
   b. Wählen Sie in der zweiten Zeile `victory_status` den Namen der Spalte aus. Lassen Sie Aggregieren auf Gruppieren nach eingestellt.
   c. Wählen Sie „Weitere Spalte hinzufügen“.
   d. Wählen Sie in der dritten Zeile `winner` den Namen der Spalte aus. Stellen Sie Aggregieren auf Anzahl ein.
   e. Wählen Sie als Gruppentyp die Option Als neue Tabelle gruppieren aus. Das Vorschaufenster zeigt Ihnen, wie das Ergebnis aussehen wird.
   f. Wählen Sie Finish (Abschließen).
6. Wählen Sie Veröffentlichen, um Ihre Arbeit zu speichern, rechts im Rezeptbereich.
7. Geben Sie als Versionsbeschreibung Erste Version meines Rezepts ein. Wählen Sie dann Veröffentlichen.

Schritt 3: Weitere Transformationen hinzufügen

In diesem Schritt fügen Sie Ihrem Rezept weitere Transformationen hinzu und veröffentlichen eine weitere Version davon. Um unser Beispiel zu verfeinern, verwenden wir die Information, dass nicht alle Schachpartien zu einem klaren Gewinner führen. Manche Partien werden unentschieden gespielt.

Um weitere Rezepttransformationen hinzuzufügen und erneut zu veröffentlichen

1. Wählen Sie in der Transformationswerkzeugleiste „Filter“, „Nach Bedingung“ und „Nicht“, um die Spiele zu entfernen, die unentschieden gespielt wurden.
2. Stellen Sie diese Optionen wie folgt ein:

   - Quellspalte - `victory_status`
   - Zustand des Filters — Ist nicht `draw`

   Um diese Transformation zu Ihrem Rezept hinzuzufügen, wählen Sie Anwenden.
3. Ändern Sie die Daten `victory_status` so, dass sie aussagekräftiger sind. Wählen Sie dazu in der Transformationssymbolleiste „Reinigen“, „Ersetzen“, „Wert oder Muster ersetzen“ aus.
4. Stellen Sie diese Optionen wie folgt ein:

   - Quellspalte - `victory_status`
   - Geben Sie die zu ersetzenden Werte an — Wert oder Muster
   - Zu ersetzender Wert - `mate`
   - Ersetze durch den Wert - `checkmate`

   Um diese Transformation zu Ihrem Rezept hinzuzufügen, wählen Sie Anwenden.
5. Wiederholen Sie den vorherigen Schritt, wechseln Sie jedoch `resign` zu`other player resigned`.
6. Wiederholen Sie den vorherigen Schritt, wechseln Sie jedoch `outoftime` zu`time ran out`.
7. Klicken Sie rechts im Rezeptbereich auf Veröffentlichen, um Ihre Arbeit zu speichern.

Schritt 4: Überprüfe deine DataBrew Ressourcen

Nachdem Sie mit einem Beispielprojekt gearbeitet haben, sehen Sie sich die DataBrew Ressourcen an, die Sie bisher erstellt haben.

Um Ihre DataBrew Ressourcen zu überprüfen

1. Wählen Sie im Navigationsbereich Datasets aus.

   Als Sie das Beispielprojekt erstellt haben, DataBrew haben Sie einen Datensatz für Sie erstellt (`chess-games`). Die Quelldatendatei wird in Amazon S3 gespeichert und hat das Microsoft Excel-Format (`chess-games.xlsx`). Die Datei enthält Metadaten von über 20.000 Schachpartien. Der `chess-games` Datensatz enthält die Informationen, die DataBrew zum Lesen der Daten in dieser Datei erforderlich sind.
2. Wählen Sie im Navigationsbereich Projekte aus.

   Sie sollten das Projekt sehen, mit dem Sie in den vorherigen Schritten gearbeitet haben (`chess-project`). In diesem Fall benötigt jedes Projekt einen Datensatz`chess-games`. Für jedes Projekt ist auch ein Rezept erforderlich, sodass Sie im Laufe der Zeit weitere Schritte zur Datentransformation hinzufügen können. Als Sie dieses Beispielprojekt erstellt DataBrew haben, haben Sie ein neues (leeres) Rezept für Sie erstellt und es an das Projekt angehängt.
3. Wählen Sie im Navigationsbereich Rezepte und in der Spalte Rezeptname die Option chess-project-recipe aus. Hier wird das Rezept angezeigt, das Sie für Ihr Projekt DataBrew erstellt haben und das Sie durch Hinzufügen von Transformationsschritten verfeinert haben.
4. Sehen Sie sich links die Rezeptversionen an, die veröffentlicht wurden. Wählen Sie eine dieser Optionen aus, um die zugehörige Registerkarte mit den Rezeptschritten aufzurufen, auf der die Rezeptdetails und Schritte für diese Version angezeigt werden.
5. Rufen Sie die Registerkarte Datenherkunft auf, auf der angezeigt wird, woher die Daten stammen und wie sie verwendet werden. Weitere Informationen erhalten Sie, wenn Sie eines der Symbole im Diagramm auswählen.

Schritt 5: Erstellen Sie ein Datenprofil

Wenn Sie an einem Projekt arbeiten, DataBrew werden Statistiken wie die Anzahl der Zeilen in der Stichprobe und die Verteilung der Einzelwerte in jeder Spalte angezeigt. Diese und viele weitere Statistiken stellen ein Profil der Stichprobe dar.

Um ein Datenprofil anzufordern, erstellen Sie einen Profiljob und führen Sie ihn aus.

Um ein Profil für einen Datensatz zu erstellen

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie auf der Registerkarte Profiljobs die Option Job erstellen aus.
3. Geben Sie als Jobname ein`chess-data-profile`.
4. Wählen Sie als Jobtyp die Option Profiljob erstellen aus.
5. Gehen Sie im Auftragseingabebereich wie folgt vor:
   - Wählen Sie für Run on die Option Dataset aus.
   - Wählen Sie Datensatz auswählen, um eine Liste der verfügbaren Datensätze anzuzeigen, und wählen Sie`chess-games`.
6. Gehen Sie im Bereich Einstellungen für die Jobausgabe wie folgt vor:
   - Wählen Sie als Dateityp JSON (JavaScript Object Notation) aus.
   - Wählen Sie S3-Standort, um eine Liste der verfügbaren Amazon S3 S3-Buckets anzuzeigen, und wählen Sie den Bucket aus, den Sie verwenden möchten. Wählen Sie dann Durchsuchen. Wählen Sie in der Ordnerliste die Option und `databrew-output` anschließend die Option Auswählen aus.
7. Wählen Sie im Bereich Zugriffsberechtigungen die Option`AwsGlueDataBrewDataAccessRole`. Dies ist eine serviceverknüpfte Rolle, mit der Sie in Ihrem Namen auf Ihre Amazon S3 S3-Buckets DataBrew zugreifen können.
8. Wählen Sie Job erstellen und ausführen. DataBrew erstellt einen Job mit Ihren Einstellungen und führt ihn dann aus.
9. Warten Sie im Bereich Verlauf der Auftragsausführung, bis sich der Auftragsstatus von `Running` zu ändert`Succeeded`.
10. Um das Profil anzuzeigen, wählen Sie PROFIL ANZEIGEN:

Das Fenster DATENSÄTZE wird angezeigt. Nehmen Sie sich etwas Zeit, um die folgenden Registerkarten zu erkunden:

- Vorschau des Datensatzes

- Überblick über das Profil
- Spaltenstatistiken
- Statistiken zur Datenherkunft

Schritt 6: Transformieren Sie den Datensatz

Bisher haben Sie Ihr Rezept nur an einer Stichprobe des Datensatzes getestet. Jetzt ist es an der Zeit, den gesamten Datensatz zu transformieren, indem Sie einen DataBrew Rezept-Job erstellen.

Wenn der Job ausgeführt wird, DataBrew wendet Ihr Rezept auf alle Daten im Datensatz an und schreibt die transformierten Daten in einen Amazon S3 S3-Bucket. Die transformierten Daten sind vom ursprünglichen Datensatz getrennt. DataBrew ändert die Quelldaten nicht.

Bevor Sie fortfahren, stellen Sie sicher, dass Ihr Konto über einen Amazon S3 S3-Bucket verfügt, in den Sie schreiben können. Erstellen Sie in diesem Bucket einen Ordner, aus dem die Job-Ausgabe erfasst werden soll DataBrew. Gehen Sie wie folgt vor, um diese Schritte durchzuführen.

Um einen S3-Bucket und einen Ordner zum Erfassen der Jobausgabe zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/databrew/.

   Wenn Ihnen bereits ein Amazon S3 S3-Bucket zur Verfügung steht und Sie Schreibberechtigungen dafür haben, überspringen Sie den nächsten Schritt.
2. Wenn Sie keinen Amazon S3 S3-Bucket haben, wählen Sie Create Bucket. Geben Sie unter Bucket-Name einen eindeutigen Namen für Ihren neuen Bucket ein. Wählen Sie Create Bucket (Bucket erstellen) aus.
3. Wählen Sie aus der Liste der Buckets den aus, den Sie verwenden möchten.
4. Wählen Sie Create folder.
5. Geben Sie `databrew-output` als Ordnername den Namen Ordner erstellen ein und wählen Sie ihn aus.

Nachdem Sie einen Amazon S3 S3-Bucket und einen Ordner für den Job erstellt haben, führen Sie Ihren Job wie folgt aus.

Um einen Rezeptjob zu erstellen und auszuführen

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie auf der Registerkarte Rezepturaufträge die Option Job erstellen aus.
3. Geben Sie als Jobname ein`chess-winner-summary`.
4. Wählen Sie als Jobtyp die Option Create a recipe job aus.
5. Gehen Sie im Auftragseingabebereich wie folgt vor:
   - Wählen Sie für Run on die Option Dataset aus.
   - Wählen Sie Datensatz auswählen, um eine Liste der verfügbaren Datensätze anzuzeigen, und wählen Sie`chess-games`.
   - Wählen Sie „Rezept auswählen“, um eine Liste der verfügbaren Rezepte anzuzeigen, und wählen Sie`chess-project-recipe`.
6. Gehen Sie im Bereich Einstellungen für die Jobausgabe wie folgt vor:
   - Dateityp — Wählen Sie CSV (kommagetrennte Werte).
   - S3-Standort — Wählen Sie dieses Feld aus, um eine Liste der verfügbaren Amazon S3 S3-Buckets anzuzeigen, und wählen Sie den Bucket aus, den Sie verwenden möchten. Wählen Sie dann Durchsuchen. Wählen Sie in der Ordnerliste die Option und `databrew-output` anschließend die Option Auswählen aus.
7. Wählen Sie im Bereich Zugriffsberechtigungen die Option`AwsGlueDataBrewDataAccessRole`. Mit dieser serviceverknüpften Rolle können Sie in Ihrem Namen DataBrew auf Ihre Amazon S3 S3-Buckets zugreifen.
8. Wählen Sie Job erstellen und ausführen. DataBrew erstellt einen Job mit Ihren Einstellungen und führt ihn dann aus.
9. Warten Sie im Bereich Verlauf der Auftragsausführung, bis sich der Auftragsstatus von `Running` zu ändert`Succeeded`.
10. Wählen Sie Output, um auf die Amazon S3 S3-Konsole zuzugreifen. Wählen Sie Ihren S3-Bucket und dann den `databrew-output` Ordner für den Zugriff auf die Job-Ausgabe aus.
11. (Optional) Wählen Sie Herunterladen, um die Datei herunterzuladen und ihren Inhalt anzusehen.

Schritt 7: (Optional) Aufräumen

Die Komplettlösung ist abgeschlossen. Sie können die von Ihnen erstellten Ressourcen DataBrew und Amazon S3 S3-Ressourcen weiterhin verwenden oder sie löschen.

So bereinigen Sie Ressourcen

1. Öffnen Sie die DataBrew Konsole https://console.aws.amazon.com/databrew/unter und wählen Sie im Navigationsbereich Projekte aus.
2. Wählen Sie Ihr Projekt aus (Beispielprojekt). Klicken Sie bei Actions auf Delete.
3. Wählen Sie im Bereich Beispielprojekt löschen die Option Angehängte Rezeptur löschen aus. Wählen Sie dann Löschen aus. Ihr Projekt wird zusammen mit dem zugehörigen Rezept und den Jobs gelöscht.
4. Wählen Sie im Navigationsbereich Datasets aus.
5. Wählen Sie Ihren Datensatz (`chess-games`) und wählen Sie unter Aktionen die Option Löschen aus.
6. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/. Löschen Sie den `databrew-output` Ordner und seinen Inhalt.

   (Optional) Wenn Sie sicher sind, dass Sie Ihren Amazon S3 S3-Bucket nicht mehr benötigen, können Sie ihn löschen.

Verbindung zu Daten herstellen mit AWS Glue DataBrew

In AWS Glue DataBrew steht ein Datensatz für Daten, die entweder aus einer Datei hochgeladen oder an einem anderen Ort gespeichert wurden. Daten können beispielsweise in Amazon S3, in einer unterstützten JDBC-Datenquelle oder einem AWS Glue Datenkatalog gespeichert werden. Wenn Sie eine Datei nicht direkt hochladen, enthält der Datensatz auch Informationen darüber DataBrew, wie Sie eine Verbindung zu den Daten herstellen DataBrew können.

Wenn Sie Ihren Datensatz erstellen (z. B.`inventory-dataset`), geben Sie die Verbindungsdetails nur einmal ein. Ab diesem Zeitpunkt DataBrew kann ich für Sie auf die zugrunde liegenden Daten zugreifen. Mit diesem Ansatz können Sie Projekte erstellen und Transformationen für Ihre Daten entwickeln, ohne sich Gedanken über Verbindungsdetails oder Dateiformate machen zu müssen.

Themen

- Unterstützte Dateitypen für Datenquellen
- Unterstützte Verbindungen für Datenquellen und Ausgaben
- Verwenden von Datensätzen in AWS Glue DataBrew
- Verbindung zu Ihren Daten herstellen
- Verbindung zu Daten in einer Textdatei herstellen mit DataBrew
- Daten in mehreren Dateien in Amazon S3 verbinden
- Datentypen
- Erweiterte Datentypen

Unterstützte Dateitypen für Datenquellen

Die folgenden Dateianforderungen gelten für Dateien, die in Amazon S3 gespeichert sind, und für Dateien, die Sie von einem lokalen Laufwerk hochladen. DataBrew unterstützt die folgenden Dateiformate: kommagetrennte Werte (CSV), Microsoft Excel, JSON, ORC und Parquet. Sie können Dateien mit einer nicht standardmäßigen Erweiterung oder ohne Erweiterung verwenden, wenn es sich bei der Datei um einen der unterstützten Dateitypen handelt.

Wenn Sie DataBrew den Dateityp nicht ableiten können, stellen Sie sicher, dass Sie den richtigen Dateityp selbst auswählen (CSV, Excel, JSON, ORC oder Parquet). Komprimierte CSV-, JSON-, ORC- und Parquet-Dateien werden unterstützt, CSV- und JSON-Dateien müssen jedoch den

Komprimierungscodec als Dateierweiterung enthalten. Wenn Sie einen Ordner importieren, müssen alle Dateien in dem Ordner denselben Dateityp haben.

Dateiformate und unterstützte Komprimierungsalgorithmen sind in der folgenden Tabelle aufgeführt.

> **Note**
>
> CSV-, Excel- und JSON-Dateien müssen mit Unicode (UTF-8) codiert werden.

Format	Dateierweiterung (optional)	Erweiterungen für komprimierte Dateien (erforderlich)

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

Microsoft Excel-Arbeitsmappe	`.xlsx`	Keine Unterstützung für Komprimierung

JSON (JSON-Dokument und JSON-Zeilen)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

Format	Dateierweiterung (optional)	Erweiterungen für komprimierte Dateien (erforderlich)
		`.snappy` `.lz4`

Unterstützte Verbindungen für Datenquellen und Ausgaben

Sie können eine Verbindung zu den folgenden Datenquellen für DataBrew Rezepturjobs herstellen. Dazu gehören alle Datenquellen, bei denen es sich nicht um DataBrew eine Datei handelt, in die Sie direkt hochladen. Die Datenquelle, die Sie verwenden, kann als Datenbank, Data Warehouse oder etwas anderes bezeichnet werden. Wir bezeichnen alle Datenanbieter als Datenquellen oder Verbindungen.

Sie können einen Datensatz mit einer der folgenden Datenquellen erstellen.

Sie können auch Amazon S3- oder JDBC-Datenbanken verwenden AWS Glue Data Catalog, die von Amazon RDS für die Ausgabe von DataBrew Rezeptjobs unterstützt werden. Amazon AppFlow und AWS Data Exchange werden nicht als Datenspeicher für die Ausgabe von DataBrew Rezeptjobs unterstützt.

- Amazon S3

  Sie können S3 verwenden, um beliebige Datenmengen zu speichern und zu schützen. Um einen Datensatz zu erstellen, geben Sie eine S3-URL an, über die Sie auf eine Datendatei zugreifen DataBrew können, zum Beispiel: `s3://your-bucket-name/inventory-data.csv`

  DataBrew kann auch alle Dateien in einem S3-Ordner lesen, was bedeutet, dass Sie einen Datensatz erstellen können, der sich über mehrere Dateien erstreckt. Geben Sie dazu eine S3-URL in dieser Form an:`s3://your-bucket-name/your-folder-name/`.

  DataBrew unterstützt nur die folgenden Amazon S3 S3-Speicherklassen: Standard, Reduced Redundancy und S3 One Zone-IA. Standard-IA DataBrew ignoriert Dateien mit anderen Speicherklassen. DataBrew ignoriert auch leere Dateien (Dateien, die 0 Byte enthalten). Weitere Informationen zu Amazon S3 S3-Speicherklassen finden Sie unter Verwenden von Amazon S3 S3-Speicherklassen im Amazon S3 S3-Konsolen-Benutzerhandbuch.

- AWS Glue Data Catalog

Sie können den Datenkatalog verwenden, um Verweise auf Daten zu definieren, die in der AWS Cloud gespeichert sind. Mit dem Datenkatalog können Sie Verbindungen zu einzelnen Tabellen in den folgenden Diensten aufbauen:

- Datenkatalog Amazon S3
- Datenkatalog Amazon Redshift
- Datenkatalog Amazon RDS
- AWS Glue

DataBrew kann auch alle Dateien in einem Amazon S3 S3-Ordner lesen, was bedeutet, dass Sie einen Datensatz erstellen können, der sich über mehrere Dateien erstreckt. Geben Sie dazu eine Amazon S3 S3-URL in dieser Form an: `s3://your-bucket-name/your-folder-name/`

Um mit verwendet werden zu können DataBrew, Amazon S3 S3-Tabellen AWS Glue Data Catalog, die in definiert sind, eine Tabelleneigenschaft namens a hinzugefügt werden`classification`, die das Format der Daten als `csv`json,`parquet`, oder und `typeOfData` als identifiziert`file`. Wenn die Tabelleneigenschaft bei der Erstellung der Tabelle nicht hinzugefügt wurde, können Sie sie über die AWS Glue Konsole hinzufügen.

DataBrew unterstützt nur die Amazon S3 S3-Speicherklassen Standard, Reduced Redundancy und S3 One Zone-IA. Standard-IA DataBrew ignoriert Dateien mit anderen Speicherklassen. DataBrew ignoriert auch leere Dateien (Dateien, die 0 Byte enthalten). Weitere Informationen zu Amazon S3 S3-Speicherklassen finden Sie unter Verwenden von Amazon S3 S3-Speicherklassen im Amazon S3 S3-Konsolen-Benutzerhandbuch.

DataBrew kann auch von anderen Konten aus auf AWS Glue Data Catalog S3-Tabellen zugreifen, wenn eine entsprechende Ressourcenrichtlinie erstellt wurde. Sie können eine Richtlinie in der AWS Glue Konsole auf der Registerkarte Einstellungen unter Datenkatalog erstellen. Im Folgenden finden Sie ein Beispiel für eine Richtlinie speziell für eine einzelne AWS-Region.

> **Warning**
>
> Dies ist eine äußerst freizügige Ressourcenrichtlinie, die `*$ACCOUNT_TO*` uneingeschränkten Zugriff auf den Datenkatalog von gewährt. `*$ACCOUNT_FROM*` In den meisten Fällen empfehlen wir, dass Sie Ihre Ressourcenrichtlinie auf bestimmte Kataloge oder Tabellen beschränken. Weitere Informationen finden Sie unter AWS Glue Ressourcenrichtlinien für die Zugriffskontrolle im AWS Glue Entwicklerhandbuch.

In einigen Fällen möchten Sie vielleicht ein Projekt erstellen oder einen Job AWS Glue DataBrew in *$ACCOUNT_TO* einer AWS Glue Data Catalog S3-Tabelle ausführen*$ACCOUNT_FROM*, die auf einen S3-Speicherort verweist, der sich ebenfalls in befindet*$ACCOUNT_FROM*. In solchen Fällen muss die IAM-Rolle, die beim Erstellen des Projekts und des Jobs verwendet wurde, über die *$ACCOUNT_TO* Berechtigung verfügen, Objekte an diesem S3-Standort aufzulisten und von *$ACCOUNT_FROM* ihnen abzurufen. Weitere Informationen finden Sie unter Kontenübergreifendem Zugriff gewähren im AWS Glue Entwicklerhandbuch.

- Mithilfe von JDBC-Treibern verbundene Daten

  Sie können einen Datensatz erstellen, indem Sie eine Verbindung zu Daten mit einem unterstützten JDBC-Treiber herstellen. Weitere Informationen finden Sie unter Verwenden von Treibern mit AWS Glue DataBrew.

  DataBrew unterstützt offiziell die folgenden Datenquellen mithilfe von Java Database Connectivity (JDBC):

  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Amazon Redshift
  - Snowflake-Konnektor für Spark

  Die Datenquellen können sich überall befinden, von wo aus Sie eine Verbindung zu ihnen herstellen können. DataBrew Diese Liste enthält nur JDBC-Verbindungen, die wir getestet haben und die wir daher unterstützen können.

  Amazon Redshift- und Snowflake Connector for Spark-Datenquellen können auf eine der folgenden Arten verbunden werden:

  - Mit einem Tabellennamen.
  - Mit einer SQL-Abfrage, die mehrere Tabellen und Operationen umfasst.

  SQL-Abfragen werden ausgeführt, wenn Sie ein Projekt oder eine Jobausführung starten.

  Um eine Verbindung zu Daten herzustellen, für die ein nicht aufgeführter JDBC-Treiber erforderlich ist, stellen Sie sicher, dass der Treiber mit JDK 8 kompatibel ist. Um den Treiber zu verwenden,

speichern Sie ihn in S3 in einem Bucket, auf den Sie mit Ihrer IAM-Rolle für zugreifen können. DataBrew Verweisen Sie dann mit Ihrem Datensatz auf die Treiberdatei. Weitere Informationen finden Sie unter Verwenden von Treibern mit AWS Glue DataBrew.

Beispielabfrage für einen SQL-based Datensatz:

```
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
```

Einschränkungen von Custom SQL

Wenn Sie eine JDBC-Verbindung verwenden, um auf Daten für einen DataBrew Datensatz zuzugreifen, sollten Sie Folgendes beachten:

- AWS Glue DataBrew validiert nicht das benutzerdefinierte SQL, das Sie im Rahmen der Datensatzerstellung angeben. Die SQL-Abfrage wird ausgeführt, wenn Sie einen Projekt- oder Joblauf starten. DataBrew nimmt die von Ihnen bereitgestellte Abfrage und übergibt sie mithilfe der standardmäßigen oder der mitgelieferten JDBC-Treiber an die Datenbank-Engine.
- Ein mit einer ungültigen Abfrage erstellter Datensatz schlägt fehl, wenn er in einem Projekt oder Job verwendet wird. Überprüfen Sie Ihre Abfrage, bevor Sie den Datensatz erstellen.
- Die Funktion „SQL validieren“ ist nur für Redshift-based Amazon-Datenquellen verfügbar.
- Wenn Sie einen Datensatz in einem Projekt verwenden möchten, beschränken Sie die Laufzeit der SQL-Abfrage auf unter drei Minuten, um ein Timeout beim Laden des Projekts zu vermeiden. Überprüfen Sie die Laufzeit der Abfrage, bevor Sie ein Projekt erstellen.

- Amazon AppFlow

Mit Amazon AppFlow können Sie Daten von Drittanbieteranwendungen Software-as-a-Service (SaaS) wie Salesforce, Zendesk, Slack und in Amazon S3 übertragen. ServiceNow Anschließend können Sie die Daten verwenden, um einen DataBrew Datensatz zu erstellen.

In Amazon AppFlow erstellen Sie eine Verbindung und einen Datenfluss für die Übertragung von Daten zwischen Ihrer Drittanbieteranwendung und einer Zielanwendung. Wenn Sie Amazon

AppFlow mit verwenden DataBrew, stellen Sie sicher, dass es sich bei der AppFlow Amazon-Zielanwendung um Amazon S3 handelt. Andere AppFlow Amazon-Zielanwendungen als Amazon S3 werden nicht in der DataBrew Konsole angezeigt. Weitere Informationen zum Übertragen von Daten aus Ihrer Drittanbieteranwendung und zum Erstellen von AppFlow Amazon-Verbindungen und -Flows finden Sie in der AppFlow Amazon-Dokumentation.

Wenn Sie auf der Registerkarte Datensätze von die Option Neuen Datensatz Connect auswählen DataBrew und auf Amazon klicken AppFlow, werden alle Flows in Amazon angezeigt AppFlow , die mit Amazon S3 als Zielanwendung konfiguriert sind. Um die Daten eines Flows für Ihren Datensatz zu verwenden, wählen Sie diesen Flow aus.

Wenn Sie AppFlow in der Konsole Flow erstellen, Flows verwalten und Details für Amazon anzeigen auswählen, wird die DataBrew AppFlow Amazon-Konsole geöffnet, sodass Sie diese Aufgaben ausführen können.

Nachdem Sie einen Datensatz von Amazon erstellt haben AppFlow, können Sie den Flow ausführen und die Details der letzten Flow-Ausführung anzeigen, wenn Sie sich die Datensatzdetails oder Jobdetails ansehen. Wenn Sie den Flow in ausführen DataBrew, wird der Datensatz in S3 aktualisiert und kann in DataBrew verwendet werden.

Die folgenden Situationen können auftreten, wenn Sie in der DataBrew Konsole einen AppFlow Amazon-Flow auswählen, um einen Datensatz zu erstellen:

- Daten wurden nicht aggregiert — Wenn der Flow-Trigger „Auf Abruf ausführen“ oder „Nach Zeitplan mit vollständiger Datenübertragung ausführen“ lautet, stellen Sie sicher, dass Sie die Daten für den Flow aggregieren, bevor Sie ihn zum Erstellen eines DataBrew Datensatzes verwenden. Beim Aggregieren des Flows werden alle Datensätze im Flow in einer einzigen Datei zusammengefasst. Für Flows mit dem Triggertyp „Nach Zeitplan ausführen mit inkrementeller Datenübertragung“ oder „Bei Ereignis ausführen“ ist keine Aggregation erforderlich. Um Daten in Amazon zu aggregieren AppFlow, wählen Sie Flow-Konfiguration bearbeiten > Zieldetails > Zusätzliche Einstellungen > Datenübertragungspräferenz.
- Flow wurde nicht ausgeführt — Wenn der Ausführungsstatus für einen Flow leer ist, bedeutet das einen der folgenden Gründe:
  - Wenn der Trigger für die Ausführung des Flows auf Anfrage ausführen lautet, wurde der Flow noch nicht ausgeführt.
  - Wenn der Trigger für die Ausführung des Flows „Bei Ereignis ausführen“ lautet, ist das auslösende Ereignis noch nicht eingetreten.

- Wenn der Trigger für die Ausführung des Flows „Nach Zeitplan ausführen“ lautet, ist noch keine geplante Ausführung erfolgt.

  Bevor Sie einen Datensatz mit einem Schema erstellen, wählen Sie Flow ausführen für diesen Flow aus.

  Weitere Informationen finden Sie unter Amazon AppFlow Flows im AppFlow Amazon-Benutzerhandbuch.

- AWS Data Exchange

  Sie können aus Hunderten von Datenquellen von Drittanbietern wählen, die in verfügbar sind AWS Data Exchange. Wenn Sie diese Datenquellen abonnieren, erhalten Sie die aktuellste Version der Daten.

  Um einen Datensatz zu erstellen, geben Sie den Namen eines AWS Data Exchange Datenprodukts an, das Sie abonniert haben und zu dessen Nutzung Sie berechtigt sind.

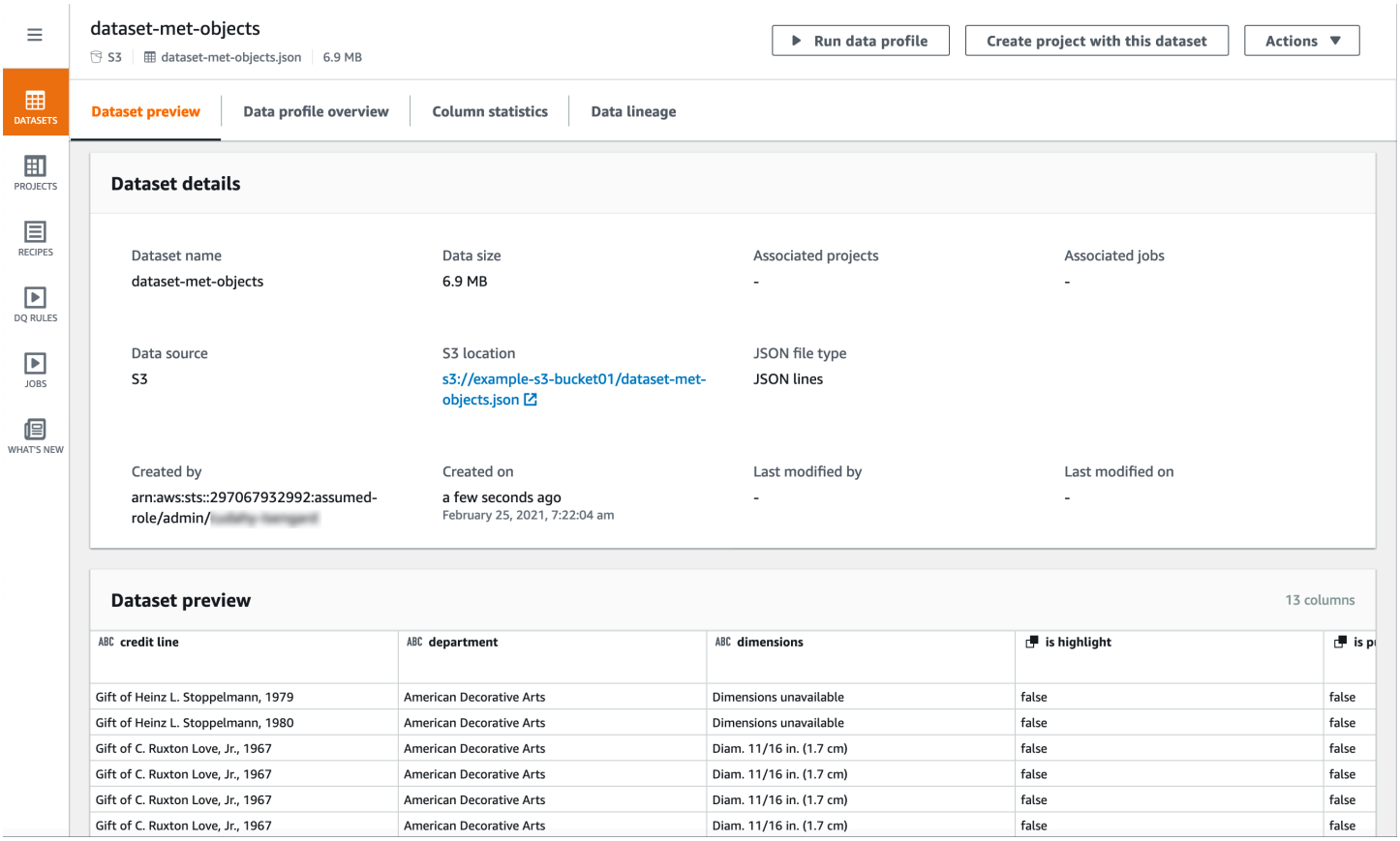
Verwenden von Datensätzen in AWS Glue DataBrew

Um eine Liste Ihrer Datensätze in der DataBrew Konsole anzuzeigen, wählen Sie auf der linken Seite DATASET aus. Auf der Datensatzseite können Sie detaillierte Informationen zu jedem Datensatz einsehen, indem Sie auf seinen Namen klicken oder im Kontextmenü Aktionen, Bearbeiten wählen.

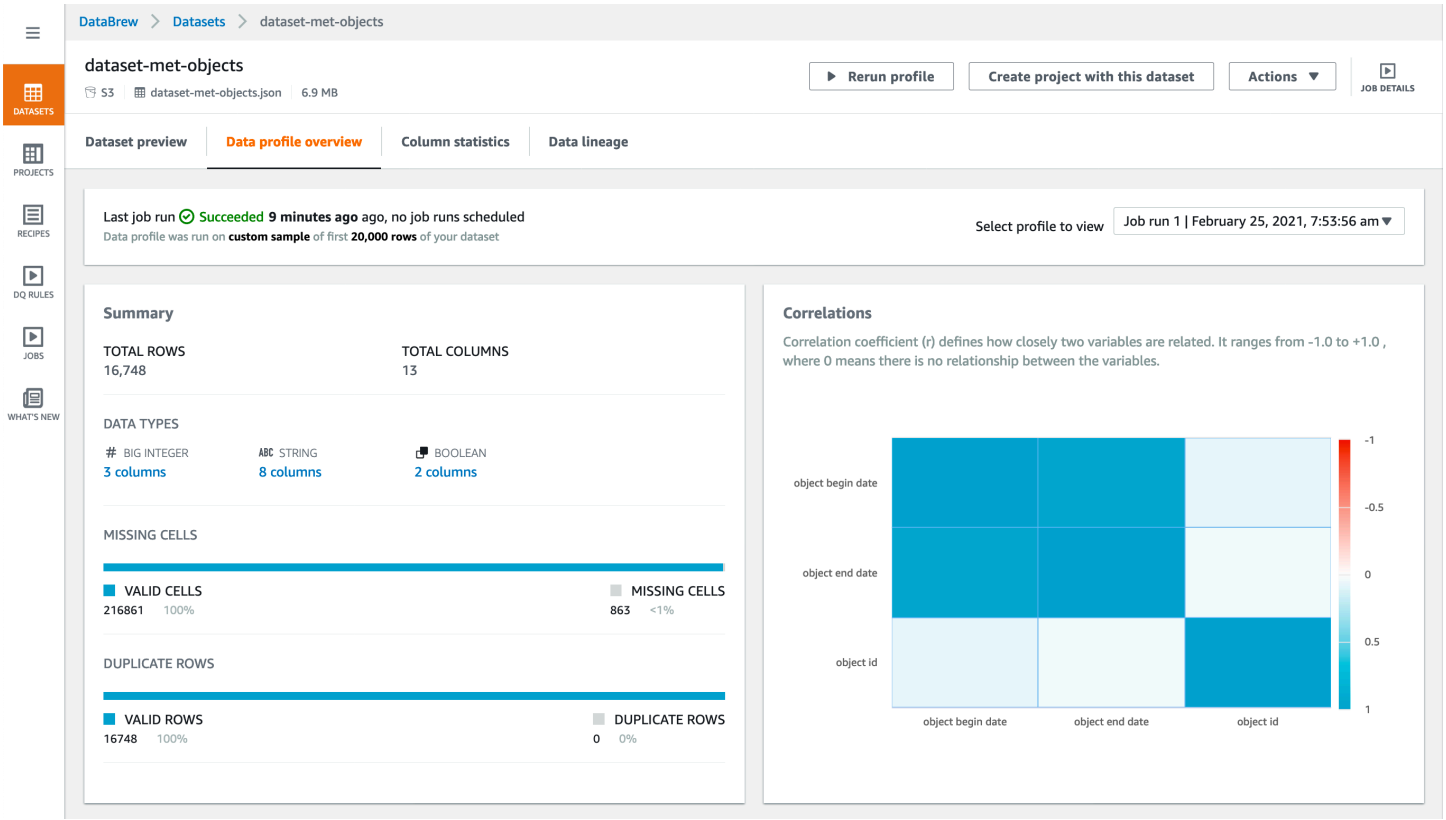
Um einen neuen Datensatz zu erstellen, wählen Sie DATENSATZ, Neuen Datensatz Connect. Verschiedene Datenquellen haben unterschiedliche Verbindungsparameter, und Sie geben diese ein, damit eine Verbindung hergestellt DataBrew werden kann. Wenn Sie Ihre Verbindung speichern und Datensatz erstellen wählen, wird eine DataBrew Verbindung zu Ihren Daten hergestellt und mit dem Laden der Daten begonnen. Weitere Informationen finden Sie unter Verbindung zu Ihren Daten herstellen.

Die Datensatzseite enthält die folgenden Elemente, die Ihnen beim Erkunden Ihrer Daten helfen sollen.

Datensatzvorschau — Auf dieser Registerkarte finden Sie Verbindungsinformationen für den Datensatz und einen Überblick über die Gesamtstruktur des Datensatzes, wie im Folgenden dargestellt.

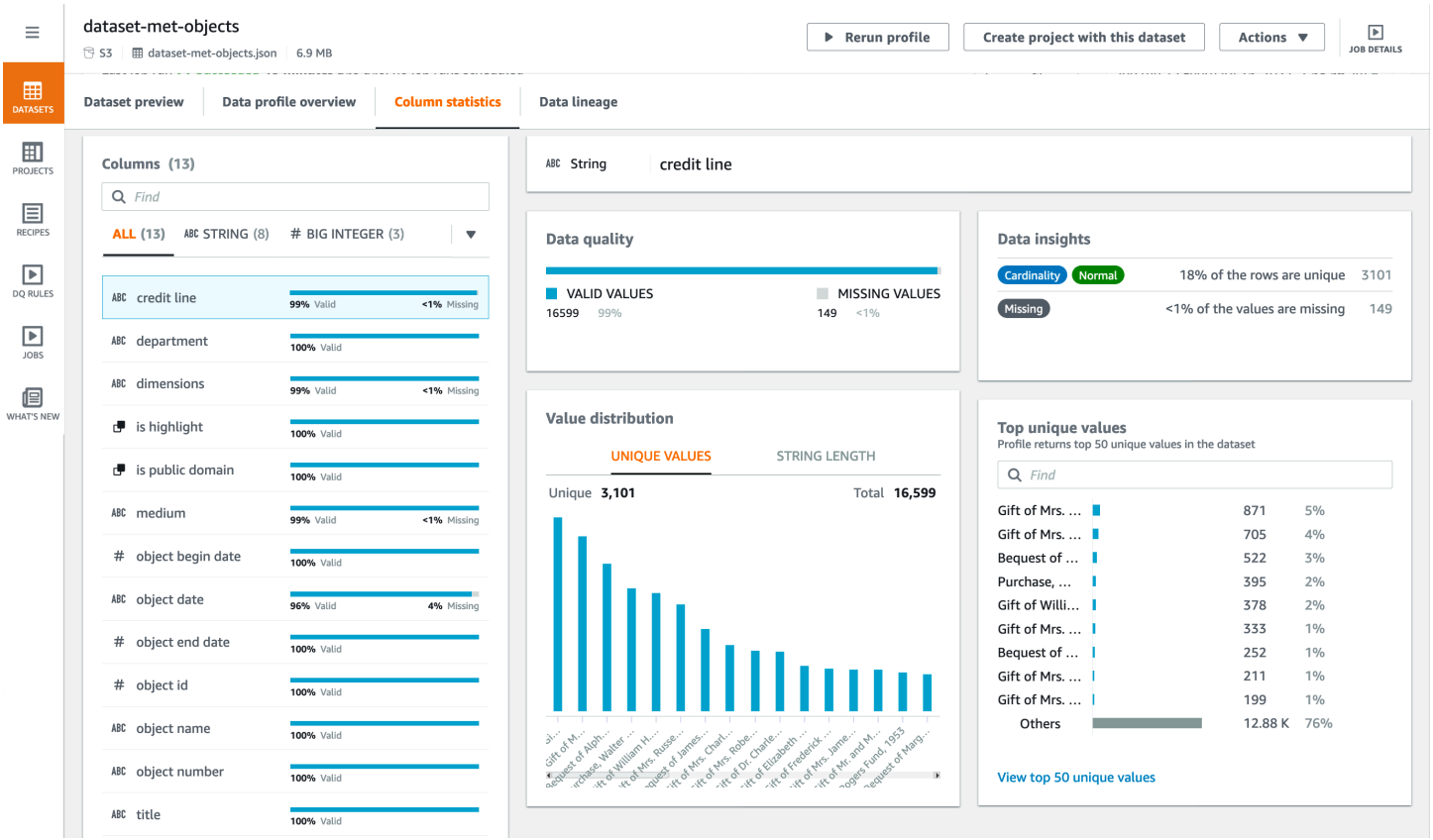


Überblick über das Datenprofil — Auf dieser Registerkarte finden Sie ein grafisches Datenprofil mit Statistiken und Volumetrie für Ihren Datensatz, wie im Folgenden dargestellt.

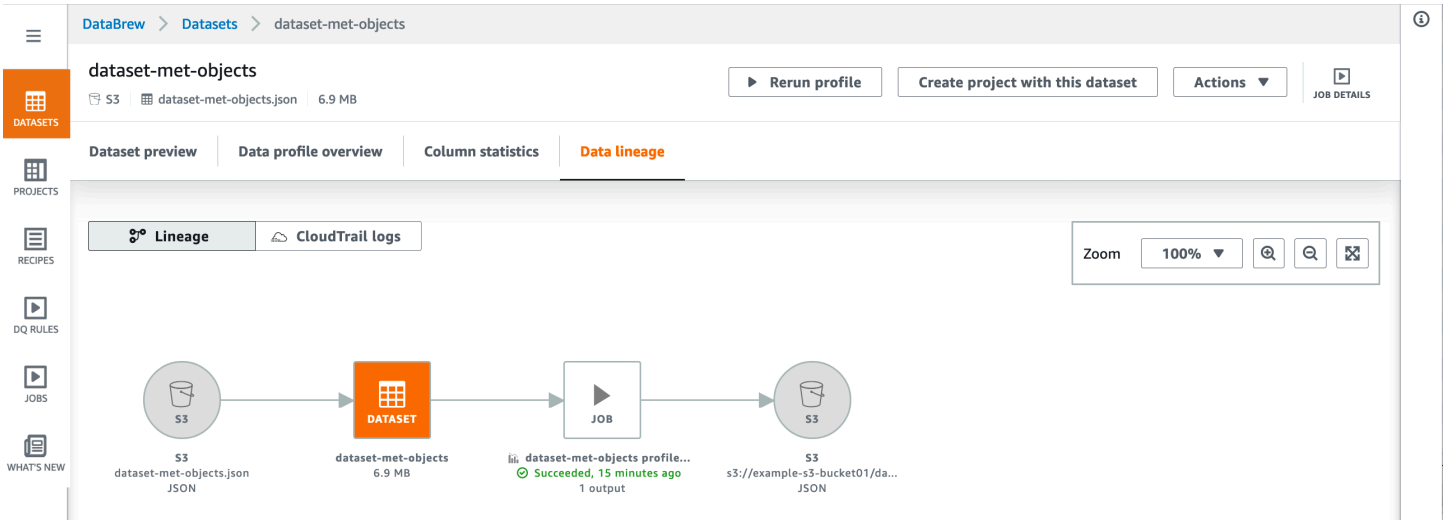


> **Note**
>
> Um ein Datenprofil zu erstellen, führen Sie einen DataBrew Profiljob für Ihren Datensatz aus. Weitere Informationen über die entsprechende Vorgehensweise finden Sie unter Schritt 5: Erstellen Sie ein Datenprofil.

Spaltenstatistiken — Auf dieser Registerkarte finden Sie detaillierte Statistiken zu jeder Spalte in Ihrem Datensatz, wie im Folgenden dargestellt.



Datenherkunft — Auf dieser Registerkarte wird grafisch dargestellt, wie Ihr Datensatz erstellt wurde und wie er verwendet wird DataBrew, wie im Folgenden dargestellt.



Themen

- Löschen eines Datensatzes

Löschen eines Datensatzes

Wenn Sie einen Datensatz nicht mehr benötigen, können Sie ihn löschen. Das Löschen eines Datensatzes hat keinerlei Auswirkungen auf die zugrunde liegende Datenquelle. Es werden lediglich die Informationen entfernt, die für den Zugriff auf die Datenquelle DataBrew verwendet wurden.

Sie können einen Datensatz nicht löschen, wenn andere DataBrew Ressourcen darauf angewiesen sind. Wenn Sie beispielsweise derzeit ein DataBrew Projekt haben, das den Datensatz verwendet, löschen Sie zuerst das Projekt, bevor Sie den Datensatz löschen.

Um einen Datensatz zu löschen, wählen Sie im Navigationsbereich Datensatz aus. Wählen Sie den Datensatz aus, den Sie löschen möchten, und wählen Sie dann für Aktionen die Option Löschen aus.

Verbindung zu Ihren Daten herstellen

Weitere Informationen zum Herstellen einer Verbindung zu den folgenden Datenquellen finden Sie in dem Abschnitt, der auf Sie zutrifft.

- AWS Glue Data Catalog— Sie können den Datenkatalog verwenden, um Verweise auf in der AWS Cloud gespeicherte Datenobjekte zu definieren, einschließlich der folgenden Dienste:
  - Amazon Redshift
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS für MySQL
  - Amazon RDS für PostgreSQL

  DataBrew erkennt alle Lake Formation Formation-Berechtigungen an, die auf Data Catalog-Ressourcen angewendet wurden, sodass DataBrew Benutzer nur dann auf diese Ressourcen zugreifen können, wenn sie autorisiert sind.

  Um einen Datensatz zu erstellen, geben Sie einen Datenkatalog-Datenbanknamen und einen Tabellennamen an. DataBrew kümmert sich um die anderen Verbindungsdetails.
- AWS Data Exchange — Sie können aus Hunderten von Datenquellen von Drittanbietern wählen, die in AWS Data Exchange verfügbar sind. Wenn Sie diese Datenquellen abonnieren, verfügen Sie immer über die aktuellste Version der Daten.

  Um einen Datensatz zu erstellen, geben Sie den Namen eines Data Exchange Exchange-Datenprodukts an, das Sie abonniert haben oder zu dessen Nutzung Sie berechtigt sind.

- JDBC-Treiberverbindungen — Sie können einen Datensatz erstellen, indem Sie eine Verbindung DataBrew zu einer JDBC-compatible Datenquelle herstellen. DataBrew unterstützt die Verbindung zu den folgenden Quellen über JDBC:
  - Amazon Redshift
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Snowflake

Themen

- Verwenden von Treibern mit AWS Glue DataBrew
- Unterstützte JDBC-Treiber

Verwenden von Treibern mit AWS Glue DataBrew

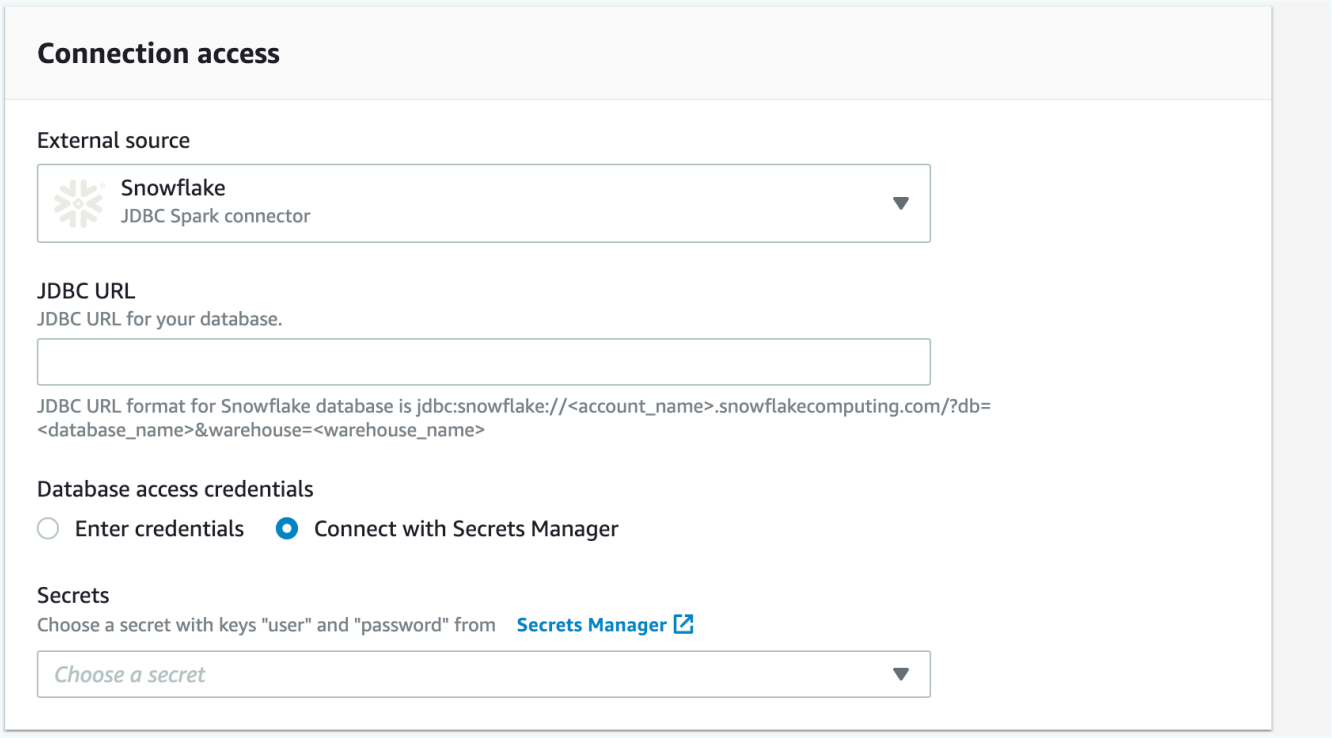
Ein Datenbanktreiber ist eine Datei oder URL, die ein Datenbankverbindungsprotokoll implementiert, beispielsweise Java Database Connectivity (JDBC). Der Treiber fungiert als Adapter oder Übersetzer zwischen einem bestimmten Datenbankmanagementsystem (DBMS) und einem anderen System.

In diesem Fall ermöglicht er AWS Glue DataBrew die Verbindung zu Ihren Daten. Anschließend können Sie von einer unterstützten Datenquelle aus auf ein Datenbankobjekt wie eine Tabelle oder Ansicht zugreifen. Die Datenquelle, die Sie verwenden, kann als Datenbank, Data Warehouse oder etwas anderes bezeichnet werden. Für die Zwecke dieser Dokumentation bezeichnen wir jedoch alle Datenanbieter als Datenquellen oder Verbindungen.

Um einen JDBC-Treiber oder eine JAR-Datei zu verwenden, laden Sie die benötigte (n) Datei (en) herunter und legen Sie sie in einen S3-Bucket. Die IAM-Rolle, die Sie für den Zugriff auf die Daten verwenden, benötigt Leseberechtigungen für beide Treiberdateien.

> **Note**
>
> With AWS Glue4.0, das Herstellen einer Verbindung zu Snowflake als Datenquelle wird nativ unterstützt. Sie müssen keine benutzerdefinierten Dateien bereitstellen. `jar` Wählen Sie AWS Glue DataBrew unter Snowflake als externe Quellverbindung aus und geben Sie die

URL Ihrer Snowflake-Instanz an. Die URL verwendet einen Hostnamen im Format `https://account_identifier.snowflakecomputing.com`.
Geben Sie die Datenzugriffsanmeldeinformationen, den Namen der Snowflake-Datenbank und den Namen des Snowflake-Schemas ein. Wenn Ihr Snowflake-Benutzer kein Standard-Warehouse-Set hat, müssen Sie außerdem einen Warehouse-Namen angeben.
Snowflake-Verbindungen verwenden ein AWS Secrets Manager Geheimnis, um Anmeldeinformationen bereitzustellen. Ihre Projekt- und Jobrollen in müssen berechtigt sein, dieses Geheimnis zu lesen.



Um Treiber zu verwenden mit DataBrew

1. Finden Sie mithilfe der vom Produkt bereitgestellten Methode heraus, welche Version Ihrer Datenquelle Sie verwenden.
2. Finden Sie die neueste Version der benötigten Konnektoren und Treiber heraus. Sie finden diese Informationen auf der Website des Datenanbieters.
3. Laden Sie die erforderliche Version der JDBC-Dateien herunter. Diese werden normalerweise als Java-Archive-Dateien (.JAR) gespeichert.
4. Laden Sie entweder die Treiber von der Konsole in Ihren S3-Bucket hoch oder geben Sie den S3-Pfad zu Ihren JAR-Dateien an.

5. Geben Sie die grundlegenden Verbindungsdetails ein, zum Beispiel Klasse, Instanz usw.
6. Geben Sie alle zusätzlichen Konfigurationsinformationen ein, die Ihre Datenquelle benötigt, z. B. Informationen zur Virtual Private Cloud (VPC).

Unterstützte JDBC-Treiber

Produkt	Unterstützte - Version	Treiberanweisungen und Downloads	SQL-Abfragen werden unterstützt
Microsoft SQL Server	v6.x oder höher	Microsoft JDBC-Treiber für SQL Server	Nicht unterstützt
MySQL	v5.1 oder höher	MySQL-Konnektoren	Nicht unterstützt
Oracle	v11.2 oder höher	Oracle JDBC wird heruntergeladen	Nicht unterstützt
PostgreSQL	v4.2.x oder höher	PostgreSQL JDBC-Treiber	Nicht unterstützt
Amazon Redshift	v4.1 oder höher	Mit JDBC eine Verbindung zu Amazon Redshift herstellen	Unterstützt
Snowflake	Um Ihre Snowflake	Um eine Verbindung zu Snowflake herzustellen, benötigen Sie die beiden folgenden Voraussetzungen:	Unterstützt

Produkt	Unterstützte - Version	Treiberanweisungen und Downloads	SQL-Abfragen werden unterstützt
	-Version zu sehen, verwende Sie CURREN ERSION, wie in der Snowflake - Dokumen ation beschrieb en.	• Snowflake-JDBC-Treiber • Snowflake-Konnektor für Spark	

Um eine Verbindung zu Datenbanken oder Data Warehouses herzustellen, die eine andere Version des Treibers benötigen als die, die DataBrew nativ unterstützt wird, können Sie einen JDBC-Treiber Ihrer Wahl bereitstellen. Der Treiber muss mit JDK 8 oder Java 8 kompatibel sein. Anweisungen, wie Sie die neueste Treiberversion für Ihre Datenbank finden, finden Sie unterVerwenden von Treibern mit AWS Glue DataBrew.

Verbindung zu Daten in einer Textdatei herstellen mit DataBrew

Sie können die folgenden Formatoptionen für die DataBrew unterstützten Eingabedateien konfigurieren:

- Comma-separated Wertdateien (CSV)
  - Trennzeichen

Das Standardtrennzeichen ist ein Komma für CSV-Dateien. Wenn Ihre Datei ein anderes Trennzeichen verwendet, wählen Sie das Trennzeichen für das CSV-Trennzeichen im Abschnitt Zusätzliche Konfigurationen aus, wenn Sie Ihren Datensatz erstellen. Die folgenden Trennzeichen werden für CSV-Dateien unterstützt:

- Komma (,)
- Doppelpunkt (:)
- Semi-colon (;)
- Pipe (|)
- Tab (\t)
- Buchstaben (^)
- Backslash (\)
- Leerzeichen

- Werte der Spaltenüberschriften

  Ihre CSV-Datei kann eine Kopfzeile als erste Zeile der Datei enthalten. Ist dies nicht der Fall, DataBrew wird eine Kopfzeile für Sie erstellt.

  - Wenn Ihre CSV-Datei eine Kopfzeile enthält, wählen Sie Erste Zeile als Kopfzeile behandeln aus. Wenn Sie dies tun, wird die erste Zeile Ihrer CSV-Datei so behandelt, als ob sie die Werte der Spaltenüberschrift enthält.
  - Wenn Ihre CSV-Datei keine Kopfzeile enthält, wählen Sie Standardüberschrift hinzufügen. Wenn Sie dies tun, DataBrew erstellt eine Kopfzeile für die Datei und behandelt Ihre erste Datenzeile nicht so, als ob sie Kopfzeilenwerte enthält. Die DataBrew erstellten Überschriften bestehen aus einem Unterstrich und einer Zahl für jede Spalte in der Datei im Format `Column_1` `Column_2``Column_3`,, usw.

- JSON-Dateien

  DataBrew unterstützt zwei Formate für JSON-Dateien: JSON Lines und JSON-Dokument. JSON Lines-Dateien enthalten eine Zeile pro Zeile. In JSON-Dokumentdateien sind alle Zeilen in einer einzigen JSON-Struktur oder einem Array enthalten. Sie können Ihren JSON-Dateityp im Abschnitt Zusätzliche Konfigurationen angeben, wenn Sie einen JSON-Datensatz erstellen. Das Standardformat ist JSON Lines.

- Excel-Dateien

  Folgendes gilt für Excel-Tabellen in DataBrew:

- Excel-Blatt wird geladen

  DataBrew Lädt standardmäßig das erste Blatt in Ihre Excel-Datei. Sie können jedoch im Abschnitt Zusätzliche Konfigurationen eine andere Blattnummer oder einen anderen Blattnamen angeben, wenn Sie einen Excel-Datensatz erstellen.

- Werte der Spaltenüberschriften

  Ihre Excel-Tabellen können eine Kopfzeile als erste Zeile der Datei enthalten. Wenn dies nicht der Fall ist, DataBrew wird eine Kopfzeile für Sie erstellt.

  - Wenn Ihre Excel-Tabellen eine Kopfzeile enthalten, wählen Sie Erste Zeile als Kopfzeile behandeln. Wenn Sie dies tun, wird die erste Zeile Ihrer Excel-Tabellen so behandelt, als ob sie die Werte der Spaltenüberschriften enthält.
  - Wenn Ihre Excel-Datei keine Kopfzeile enthält, wählen Sie Standardüberschrift hinzufügen. Auf diese Weise geben Sie an, dass eine Kopfzeile für die Datei erstellt werden DataBrew soll und Ihre erste Datenzeile nicht so behandelt werden soll, als ob sie Kopfzeilenwerte enthält. Die Kopfzeilen, die DataBrew erstellt werden, bestehen aus einem Unterstrich und einer Zahl für jede Spalte in der Datei im Format `Column_1` `Column_2Column_3`,, usw.

Daten in mehreren Dateien in Amazon S3 verbinden

Mit der DataBrew Konsole können Sie in Amazon S3 S3-Buckets und -Ordnern navigieren und eine Datei für Ihren Datensatz auswählen. Ein Datensatz muss jedoch nicht auf eine Datei beschränkt sein.

Angenommen, Sie haben einen S3-Bucket mit dem Namen`my-databrew-bucket`, der einen Ordner mit dem Namen enthält`databrew-input`. Nehmen wir an, Sie haben in diesem Ordner eine Reihe von JSON-Dateien, die alle dasselbe Dateiformat und dieselbe `.json` Dateierweiterung haben. Auf der Konsole können Sie eine Quell-URL von angeben`s3://my-databrew-bucket/databrew-input/`. Auf der DataBrew Konsole können Sie dann diesen Ordner auswählen. Ihr Datensatz besteht aus allen JSON-Dateien in diesem Ordner.

DataBrew kann alle Dateien in einem S3-Ordner verarbeiten, aber nur, wenn die folgenden Bedingungen zutreffen:

- Alle Dateien im Ordner haben dasselbe Format.
- Alle Dateien im Ordner haben dieselbe Dateierweiterung.

Weitere Informationen zu unterstützten Dateiformaten und Erweiterungen finden Sie unterDataBrew input formats.

Schemas bei der Verwendung mehrerer Dateien als Datensatz

Wenn Sie mehrere Dateien als DataBrew Datensatz verwenden, müssen die Schemas für alle Dateien identisch sein. Andernfalls versucht der Projektarbeitsbereich automatisch, eines der Schemas aus den mehreren Dateien auszuwählen, und versucht, die restlichen Datensatzdateien an dieses Schema anzupassen. Dieses Verhalten führt dazu, dass die Ansicht, die in Project Workspace angezeigt wird, unregelmäßig ist, sodass auch die Jobausgabe unregelmäßig ist.

Wenn Ihre Dateien unterschiedliche Schemas haben müssen, müssen Sie mehrere Datensätze erstellen und diese separat profilieren.

Verwenden von parametrisierten Pfaden für Amazon S3

In einigen Fällen möchten Sie vielleicht einen Datensatz mit Dateien erstellen, die einer bestimmten Namenskonvention folgen, oder einen Datensatz, der sich über mehrere Amazon S3 S3-Ordner erstrecken kann. Oder vielleicht möchten Sie denselben Datensatz für identisch strukturierte Daten wiederverwenden, die regelmäßig an einem S3-Speicherort generiert werden, deren Pfad von bestimmten Parametern abhängt. Ein Beispiel ist ein Pfad, der nach dem Datum der Datenproduktion benannt ist.

DataBrew unterstützt diesen Ansatz mit parametrisierten S3-Pfaden. Ein parametrisierter Pfad ist eine Amazon S3 S3-URL, die reguläre Ausdrücke oder benutzerdefinierte Pfadparameter oder beides enthält.

Definition eines Datensatzes mit einem S3-Pfad unter Verwendung regulärer Ausdrücke

Reguläre Ausdrücke im Pfad können nützlich sein, um mehrere Dateien aus einem oder mehreren Ordnern zuzuordnen und gleichzeitig nicht verwandte Dateien in diesen Ordnern herauszufiltern.

Hier sind ein paar Beispiele:

- Definieren Sie einen Datensatz, der alle JSON-Dateien aus einem Ordner enthält, dessen Name mit beginnt`invoice`.
- Definieren Sie einen Datensatz, der alle Dateien in Ordnern mit `2020` ihren Namen enthält.

Sie können diese Art von Ansatz implementieren, indem Sie reguläre Ausdrücke in einem S3-Pfad eines Datensatzes verwenden. Diese regulären Ausdrücke können jede Teilzeichenfolge im Schlüssel der S3-URL ersetzen (aber nicht den Bucket-Namen).

Ein Beispiel für einen Schlüssel in einer S3-URL finden Sie im Folgenden. Hier `my-bucket` ist der Bucket-Name, US East (Ohio) ist die AWS Region und `puppy.png` der Schlüsselname.

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

In einem parametrisierten S3-Pfad werden alle Zeichen zwischen zwei spitzen Klammern (<und>) als reguläre Ausdrücke behandelt. Zwei Beispiele sind die folgenden:

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json`entspricht allen benannten `data.json` Dateien in allen Unterordnern`databrew-input`, deren Namen mit `invoice` beginnen.
- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/`stimmt mit allen Dateien in Ordnern überein, die `2020` in ihren Namen vorkommen.

Entspricht in diesen `.*` Beispielen null oder mehr Zeichen.

> **Note**
>
> Sie können reguläre Ausdrücke nur im Schlüsselteil des S3-Pfads verwenden — dem Teil, der nach dem Bucket-Namen steht. Ist also gültig, `s3://my-databrew-bucket/<.*>-input/` ist es aber `s3://my-<.*>-bucket/<.*>-input/` nicht.

Wir empfehlen Ihnen, Ihre regulären Ausdrücke zu testen, um sicherzustellen, dass sie nur den gewünschten S3-URLs entsprechen und nicht den URLs, die Sie nicht möchten.

Hier sind einige andere Beispiele für reguläre Ausdrücke:

- `<\d{2}>`entspricht einer Zeichenfolge, die aus genau zwei aufeinanderfolgenden Ziffern besteht, zum Beispiel `07` oder`03`, aber nicht`1a2`.
- `<[a-z]+.*>`entspricht einer Zeichenfolge, die mit einem oder mehreren lateinischen Kleinbuchstaben beginnt und nach der kein oder mehrere weitere Zeichen folgen. Ein Beispiel ist`a3`, oder `abc/defa-z`, aber nicht`A2`.
- `<[^/]+>`entspricht einer Zeichenfolge, die beliebige Zeichen außer einem Schrägstrich (/) enthält. In einer S3-URL werden Schrägstriche verwendet, um Ordner im Pfad zu trennen.

- `<.*=.*>`entspricht einer Zeichenfolge, die ein Gleichheitszeichen (=) enthält, z. B.`month=02`, oder `abc/day=2=10`, aber nicht. `test`
- `<\d.*\d>`entspricht einer Zeichenfolge, die mit einer Ziffer beginnt und endet und zwischen den Ziffern beliebige andere Zeichen enthalten kann, z. B.`1abc2`, oder `01-02-032020/Jul/21`, aber nicht. `123a`

Definieren eines Datensatzes mit einem S3-Pfad mithilfe benutzerdefinierter Parameter

Die Definition eines parametrisierten Datensatzes mit benutzerdefinierten Parametern bietet Vorteile gegenüber der Verwendung regulärer Ausdrücke, wenn Sie möglicherweise Parameter für einen S3-Standort angeben möchten:

- Sie können dieselben Ergebnisse wie mit einem regulären Ausdruck erzielen, ohne die Syntax für reguläre Ausdrücke kennen zu müssen. Sie können Parameter mit vertrauten Begriffen wie „beginnt mit“ und „enthält“ definieren.
- Wenn Sie einen dynamischen Datensatz mithilfe von Parametern im Pfad definieren, können Sie einen Zeitraum in Ihre Definition aufnehmen, z. B. „letzter Monat“ oder „letzte 24 Stunden“. Auf diese Weise wird Ihre Datensatzdefinition später für neue eingehende Daten verwendet.

Hier sind einige Beispiele dafür, wann Sie dynamische Datensätze verwenden sollten:

- Um mehrere Dateien, die nach dem Datum der letzten Aktualisierung oder anderen aussagekräftigen Attributen partitioniert sind, zu einem einzigen Datensatz zu verbinden. Sie können diese Partitionsattribute dann als zusätzliche Spalten in einem Datensatz erfassen.
- Um Dateien in einem Datensatz auf S3-Speicherorte zu beschränken, die bestimmte Bedingungen erfüllen. Nehmen wir zum Beispiel an, dass Ihr S3-Pfad datumsbasierte Ordner wie enthält. `folder/2021/04/01/` In diesem Fall können Sie das Datum parametrisieren und es auf einen bestimmten Bereich beschränken, z. B. „zwischen dem 01. März 2021 und dem 01. April 2021“ oder „Letzte Woche“.

Um einen Pfad mithilfe von Parametern zu definieren, definieren Sie die Parameter und fügen Sie sie Ihrem Pfad im folgenden Format hinzu:

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

> **Note**
>
> Wie bei regulären Ausdrücken in einem S3-Pfad können Sie Parameter nur im Schlüsselteil des Pfads verwenden — dem Teil, der nach dem Bucket-Namen steht.

In einer Parameterdefinition sind zwei Felder erforderlich: Name und Typ. Der Typ kann Zeichenfolge, Zahl oder Datum sein. Parameter des Typs Datum müssen eine Definition des Datumsformats haben, damit Datumswerte korrekt interpretiert und verglichen werden DataBrew können. Optional können Sie Übereinstimmungsbedingungen für einen Parameter definieren. Sie können sich auch dafür entscheiden, passende Werte eines Parameters als Spalte zu Ihrem Datensatz hinzuzufügen, wenn dieser durch einen DataBrew Job oder eine interaktive Sitzung geladen wird.

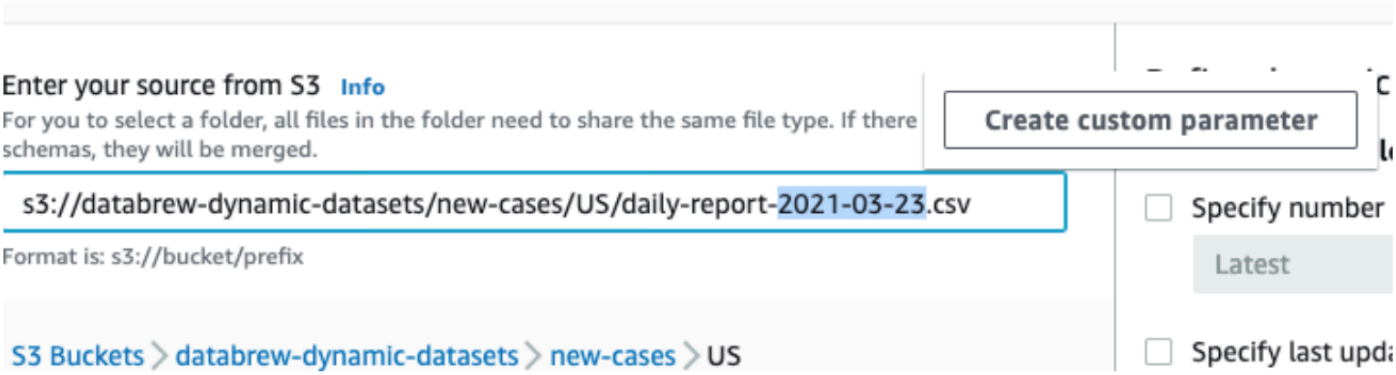
Beispiel

Betrachten wir ein Beispiel für die Definition eines dynamischen Datensatzes mithilfe von Parametern in der DataBrew Konsole. Gehen Sie in diesem Beispiel davon aus, dass die Eingabedaten regelmäßig unter Verwendung von Speicherorten wie diesen in einen S3-Bucket geschrieben werden:

- `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`

Hier gibt es zwei dynamische Teile: einen Ländercode, wie in den USA, und ein Datum im Dateinamen wie 2021-03-30. Hier können Sie dasselbe Bereinigungsrezept für alle Dateien anwenden. Nehmen wir an, Sie möchten Ihre Bereinigungsaufgabe täglich durchführen. Im Folgenden erfahren Sie, wie Sie einen parametrisierten Pfad für dieses Szenario definieren können:

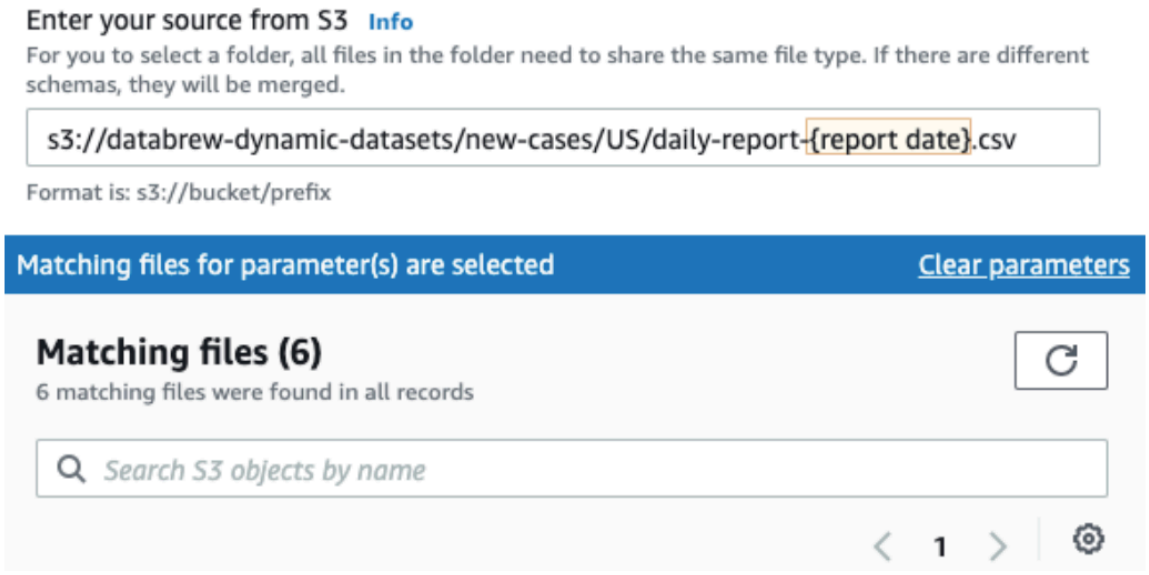
1. Navigieren Sie zu einer bestimmten Datei.
2. Wählen Sie dann einen variierenden Teil aus, z. B. ein Datum, und ersetzen Sie ihn durch einen Parameter. Ersetzen Sie in diesem Fall ein Datum.



3. Öffnen Sie das Kontextmenü (Rechtsklick) für Benutzerdefinierten Parameter erstellen und legen Sie dessen Eigenschaften fest:
   - Name: Berichtsdatum
   - Typ: Datum
   - Datumsformat: yyyy- MM-dd (ausgewählt aus den vordefinierten Formaten)
   - Bedingungen (Zeitraum): Letzte 24 Stunden
   - Als Spalte hinzufügen: true (aktiviert)

   Behalten Sie die Standardwerte für andere Felder bei.
4. Wählen Sie Erstellen aus.

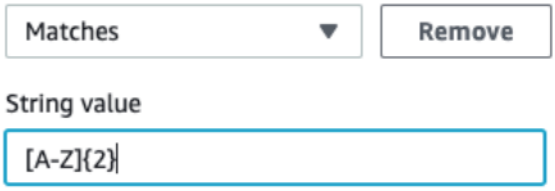
Danach sehen Sie den aktualisierten Pfad wie im folgenden Screenshot.



Jetzt können Sie dasselbe für den Ländercode tun und ihn wie folgt parametrisieren:

- Name: Landesvorwahl
- Typ: Zeichenfolge
- Als Spalte hinzufügen: wahr (aktiviert)

Sie müssen keine Bedingungen angeben, wenn alle Werte relevant sind. In dem `new-cases` Ordner haben wir beispielsweise nur Unterordner mit Länderkennzahlen, sodass keine Bedingungen erforderlich sind. Wenn Sie andere Ordner ausschließen möchten, könnten Sie die folgende Bedingung verwenden.



Bei diesem Ansatz werden die Unterordner neuer Fälle auf zwei lateinische Großbuchstaben beschränkt.

Nach dieser Parametrisierung haben Sie nur noch passende Dateien in unserem Datensatz und können Datensatz erstellen wählen.

> **Note**
>
> Wenn Sie relative Zeitbereiche in Bedingungen verwenden, werden die Zeitbereiche beim Laden des Datensatzes ausgewertet. Dies gilt unabhängig davon, ob es sich um vordefinierte Zeitbereiche wie „Letzte 24 Stunden“ oder um benutzerdefinierte Zeitbereiche wie „vor 5 Tagen“ handelt. Dieser Bewertungsansatz gilt unabhängig davon, ob der Datensatz während der Initialisierung einer interaktiven Sitzung oder während eines Jobstarts geladen wird.

Nachdem Sie „Datensatz erstellen“ ausgewählt haben, ist Ihr dynamischer Datensatz einsatzbereit. Sie können es beispielsweise zunächst verwenden, um ein Projekt zu erstellen und in einer interaktiven DataBrew Sitzung ein Bereinigungsrezept zu definieren. Dann könnten Sie einen Job erstellen, der für die tägliche Ausführung geplant ist. Bei diesem Job wird das Bereinigungsrezept möglicherweise auf die Datensatzdateien angewendet, die zu dem Zeitpunkt, zu dem der Job gestartet wird, die Bedingungen Ihrer Parameter erfüllen.

Unterstützte Bedingungen für dynamische Datensätze

Sie können Bedingungen verwenden, um übereinstimmende S3-Dateien anhand von Parametern oder dem Datumsattribut der letzten Änderung zu filtern.

Im Folgenden finden Sie eine Liste der unterstützten Bedingungen für jeden Parametertyp.

Bedingungen, die mit Zeichenkettenparametern verwendet werden

Name im DataBrew SDK	SDK-Synonyme	Name in der DataBrew Konsole	Description
ist	eq, ==	Ist genau	Der Wert des Parameters entsprich t dem Wert, der in der Bedingung angegeben wurde.
ist nicht	nicht eq,! =	Ist nicht	Der Wert des Parameters entsprich t nicht dem Wert, der in der Bedingung angegeben wurde.
enthält		Enthält	Der Zeichenke ttenwert des Parameters enthält den Wert, der in der Bedingung bereitges tellt wurde.
nicht enthält		Enthält nicht	Der Zeichenke ttenwert des Parameters enthält nicht den Wert, der in der Bedingung angegeben wurde.
starts_with		Beginnt mit	Der Zeichenke ttenwert des

Name im DataBrew SDK	SDK-Synonyme	Name in der DataBrew Konsole	Description
			Parameters beginnt mit dem Wert, der in der Bedingung angegeben wurde.
nicht starts_with		Beginnt nicht mit	Der Zeichenkettenwert des Parameters beginnt nicht mit dem Wert, der in der Bedingung angegeben wurde.
ends_with		Endet mit	Der Zeichenkettenwert des Parameters endet mit dem Wert, der in der Bedingung angegeben wurde.
nicht ends_with		Endet nicht mit	Der Zeichenkettenwert des Parameters endet nicht mit dem Wert, der in der Bedingung angegeben wurde.
Streichhölzer		Entspricht	Der Wert des Parameters entspricht dem in der Bedingung angegebenen regulären Ausdruck.

Name im DataBrew SDK	SDK-Synonyme	Name in der DataBrew Konsole	Description
stimmt nicht überein		Stimmt nicht überein	Der Wert des Parameters entspricht nicht dem in der Bedingung angegebenen regulären Ausdruck.

> **Note**
>
> Bei allen Bedingungen für String-Parameter wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie sich nicht sicher sind, welche Groß- und Kleinschreibung in einem S3-Pfad verwendet wird, können Sie die Bedingung „entspricht“ mit einem regulären Ausdruckswert verwenden, der mit `(?i)` beginnt. Dies führt zu einem Vergleich ohne Berücksichtigung der Groß- und Kleinschreibung.
> Nehmen wir zum Beispiel an, dass Ihr Zeichenkettenparameter mit`abc`, aber auch mit `Abc` oder `ABC` beginnen soll. In diesem Fall können Sie die Bedingung „entspricht“ `(?i)^abc` als Bedingungswert verwenden.

Bedingungen, die mit Zahlenparametern verwendet werden

Name im DataBrew SDK	SDK-Synonyme	Name in der DataBrew Konsole	Description
ist	eq, ==	Ist genau	Der Wert des Parameters entspricht dem Wert, der in der Bedingung angegeben wurde.
ist nicht	nicht eq,! =	Ist nicht	Der Wert des Parameters entspricht nicht dem Wert, der

Name im DataBrew SDK	SDK-Synonyme	Name in der DataBrew Konsole	Description
			in der Bedingung angegeben wurde.
kleiner_als	lt, <	Kleiner als	Der numerische Wert des Parameters ist kleiner als der Wert, der in der Bedingung angegeben wurde.
less_than_equal	spät, <=	Kleiner als oder gleich	Der numerische Wert des Parameter s ist kleiner oder gleich dem Wert, der in der Bedingung angegeben wurde.
größer_als	gt, >	Größer als	Der numerische Wert des Parameters ist größer als der Wert, der in der Bedingung angegeben wurde.
greater_than_equal	erhalten, =>	Größer als oder gleich	Der numerische Wert des Parameter s ist größer oder gleich dem Wert, der in der Bedingung angegeben wurde.

Mit Date-Parametern verwendete Bedingungen

Name im DataBrew SDK	Name in der DataBrew Konsole	Format für Bedingungswerte (SDK)	Description
after	Starten	ISO 8601-Datumsformat wie oder `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	Der Wert des Datumsparameters liegt nach dem in der Bedingung angegebenen Datum.
before	Ende	ISO 8601-Datumsformat wie oder `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	Der Wert des Datumsparameters liegt vor dem in der Bedingung angegebenen Datum.
relative_after	Start (relativ)	Positive oder negative Anzahl von Zeiteinheiten, wie `-48h` oder`+7d`.	Der Wert des Datumsparameters liegt nach dem in der Bedingung angegebenen relativen Datum. Relative Daten werden ausgewertet, wenn der Datensatz geladen wird, entweder wenn eine interaktive Sitzung initialisiert wird oder wenn ein zugehöriger Job gestartet wird. Dies ist der Moment, der in den Beispielen „jetzt“ genannt wird.

Name im DataBrew SDK	Name in der DataBrew Konsole	Format für Bedingungswerte (SDK)	Description
relative_before	Ende (relativ)	Positive oder negative Anzahl von Zeiteinheiten, wie `-48h` oder`+7d`.	Der Wert des Datumsparameters liegt vor dem in der Bedingung angegebenen relativen Datum. Relative Daten werden ausgewertet, wenn der Datensatz geladen wird, entweder wenn eine interaktive Sitzung initialisiert wird oder wenn ein zugehöriger Job gestartet wird. Dies ist der Moment, der in den Beispielen „jetzt“ genannt wird.

Wenn Sie das SDK verwenden, geben Sie relative Daten im folgenden Format an:`±{number_of_time_units}{time_unit}`. Sie können diese Zeiteinheiten verwenden:

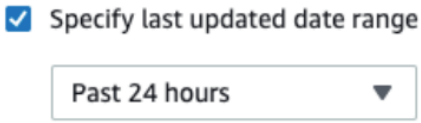
- -1h (vor 1 Stunde)
- +2d (in 2 Tagen)
- -120m (vor 120 Minuten)
- 5000s (in 5.000 Sekunden von jetzt an)
- -3w (vor 3 Wochen)
- +4M (in 4 Monaten)
- -1y (vor 1 Jahr)

Relative Daten werden ausgewertet, wenn der Datensatz geladen wird, entweder wenn eine interaktive Sitzung initialisiert wird oder wenn ein zugehöriger Job gestartet wird. Dies ist der Moment, der in den vorangegangenen Beispielen „jetzt“ genannt wird.

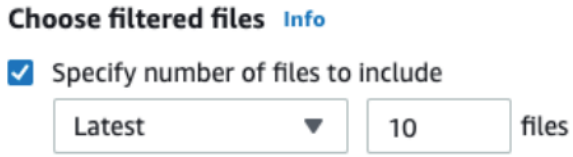
Einstellungen für dynamische Datensätze konfigurieren

Neben der Bereitstellung eines parametrisierten S3-Pfads können Sie auch andere Einstellungen für Datensätze mit mehreren Dateien konfigurieren. Diese Einstellungen filtern S3-Dateien nach ihrem letzten Änderungsdatum und begrenzen die Anzahl der Dateien.

Ähnlich wie beim Einstellen eines Datumsparameters in einem Pfad können Sie einen Zeitraum definieren, in dem passende Dateien aktualisiert wurden, und nur diese Dateien in Ihren Datensatz aufnehmen. Sie können diese Bereiche entweder mit absoluten Daten wie „30. März 2021“ oder mit relativen Zeiträumen wie „Letzte 24 Stunden“ definieren.



Um die Anzahl der passenden Dateien zu begrenzen, wählen Sie eine Anzahl von Dateien aus, die größer als 0 ist, und wählen Sie aus, ob Sie die neuesten oder die ältesten übereinstimmenden Dateien verwenden möchten.



Datentypen

Die Daten für jede Spalte Ihres Datensatzes werden in einen der folgenden Datentypen konvertiert:

- Byte — 1-Byte-Ganzzahlen mit Vorzeichen. Der Zahlenbereich reicht von -128 bis 127.
- kurz — 2-Byte-Ganzzahlen mit Vorzeichen. Der Zahlenbereich reicht von -32768 bis 32767.
- Ganzzahl — 4-Byte-Ganzzahlzahlen mit Vorzeichen. Der Zahlenbereich reicht von -2147483648 bis 2147483647.
- lang — 8-Byte-Ganzzahlzahlen mit Vorzeichen. Der Zahlenbereich reicht von -9223372036854775808 bis 9223372036854775807.
- float — 4-Byte-Gleitkommazahlen mit einfacher Genauigkeit.

- double — 8-Byte-Gleitkommazahlen mit doppelter Genauigkeit.
- Dezimalzahlen — Vorzeichenbehaftete Dezimalzahlen mit insgesamt bis zu 38 Ziffern und 18 Nachkommastellen.
- string — Zeichenkettenwerte.
- boolean — Der boolesche Typ hat einen von zwei möglichen Werten: `true` und `false` oder `yes` und `no`.
- timestamp — Werte, die die Felder Jahr, Monat, Tag, Stunde, Minute und Sekunde umfassen.
- Datum — Werte, die die Felder Jahr, Monat und Tag umfassen.

Fortgeschrittene Datentypen

Erweiterte Datentypen sind Datentypen, die innerhalb einer Zeichenkettenspalte in einem Projekt DataBrew erkannt werden und daher nicht Teil eines Datensatzes sind. Informationen zu erweiterten Datentypen finden Sie unter Erweiterte Datentypen.

Erweiterte Datentypen

Fortgeschrittene Datentypen sind Datentypen, die innerhalb einer Zeichenkettenspalte in einem Projekt mithilfe von Musterabgleich DataBrew erkannt werden. Wenn Sie auf eine Zeichenfolgenspalte klicken, wird die Spalte als der entsprechende erweiterte Datentyp gekennzeichnet, wenn mindestens 50% der Werte in der Spalte die Kriterien für diesen Datentyp erfüllen.

Folgende Datentypen DataBrew können erkannt werden:

- Date/timestamp
- SSN
- Phone number (Telefonnummer)
- Email
- Kreditkarte
- Gender
- IP-Adresse
- URL
- PLZ

- Land
- Währung
- Status
- Ort

Sie können die folgenden Transformationen verwenden, um mit erweiterten Datentypen zu arbeiten:

- GET_ADVANCED_DATATYPE: Identifiziert anhand einer Zeichenkettenspalte den erweiterten Datentyp der Spalte, falls vorhanden.
- EXTRACT_ADVANCED_DATATYPE_DETAILS: Extrahiert Details für einen erweiterten Datentyp.
- ADVANCED_DATATYPE_FILTER: Filtert eine aktuelle Quellspalte auf der Grundlage der erweiterten Datentyperkennung.
- ADVANCED_DATATYPE_FLAG: Erstellt eine neue Flaggenspalte auf der Grundlage der Werte für die aktuelle Quellspalte.

Validierung der Datenqualität in AWS Glue DataBrew

Um die Qualität Ihrer Datensätze sicherzustellen, können Sie eine Liste von Datenqualitätsregeln in einem Regelsatz definieren. Ein Regelsatz ist ein Regelsatz, der verschiedene Datenmetriken mit erwarteten Werten vergleicht. Wenn eines der Kriterien einer Regel nicht erfüllt ist, schlägt die Überprüfung des gesamten Regelsatzes fehl. Sie können dann die einzelnen Ergebnisse für jede Regel überprüfen. Für jede Regel, die zu einem Validierungsfehler führt, können Sie die erforderlichen Korrekturen vornehmen und die Überprüfung erneut durchführen.

Zu den Regeln gehören beispielsweise die folgenden:

- Der Wert in der Spalte `"APY"` liegt zwischen 0 und 100
- Die Anzahl der fehlenden Werte in der Spalte überschreitet `group_name` nicht 5%

Sie können jede Regel für eine einzelne Spalte definieren oder sie unabhängig voneinander auf mehrere ausgewählte Spalten anwenden, zum Beispiel:

- Der Maximalwert für Spalten`"rate"`,`"pay"`, überschreitet nicht 100`"increase"`.

Eine Regel kann aus mehreren einfachen Prüfungen bestehen. Sie können definieren, ob alle wahr oder beliebig sein sollen, zum Beispiel:

- Der Wert in der Spalte `"ProductId"` sollte mit `"asin-"` UND beginnen. Die Länge des Werts in der Spalte `"ProductId"` ist 32.

Sie können Regeln entweder anhand von Aggregatwerten wie`max`,`min`, oder anhand `number of duplicate values` derer nur ein Wert verglichen wird, oder anhand von nicht aggregierten Werten in jeder Zeile einer Spalte überprüfen. In letzterem Fall können Sie auch einen Schwellenwert definieren, bei dem die Anforderungen erfüllt werden, z. `value in columnA > value in columnB for at least 95% of rows`

Wie bei Profilinformationen können Sie Datenqualitätsregeln auf Spaltenebene nur für Spalten einfacher Typen wie Zeichenfolgen und Zahlen definieren. Sie können keine Datenqualitätsregeln für Spalten komplexer Typen wie Arrays oder Strukturen definieren. Weitere Informationen zum Arbeiten mit Profilinformationen finden Sie unterErstellen und Arbeiten mit AWS Glue DataBrew Jobs profilieren.

Datenqualitätsregeln validieren

Nachdem ein Regelsatz definiert wurde, können Sie ihn zur Überprüfung zu einem Profiljob hinzufügen. Sie können mehr als einen Regelsatz für einen Datensatz definieren.

Beispielsweise kann ein Regelsatz Regeln mit minimal akzeptablen Kriterien enthalten. Wenn die Überprüfung für diesen Regelsatz fehlschlägt, kann dies bedeuten, dass die Daten für die weitere Verwendung nicht akzeptabel sind. Ein Beispiel sind fehlende Werte in Schlüsselspalten eines Datensatzes, der für Schulungen zum maschinellen Lernen verwendet wird. Sie können einen zweiten Regelsatz mit strengeren Regeln verwenden, um zu überprüfen, ob der Datensatz eine so gute Qualität aufweist, dass keine Bereinigung erforderlich ist.

Sie können einen oder mehrere Regelsätze anwenden, die für einen bestimmten Datensatz in einer Profiljob-Konfiguration definiert sind. Wenn der Profiljob ausgeführt wird, erstellt er zusätzlich zum Datenprofil einen Validierungsbericht. Der Validierungsbericht ist am selben Ort wie Ihre Profildaten verfügbar. Wie bei den Profilinformationen können Sie sich die Ergebnisse in der DataBrew Konsole ansehen. Wählen Sie in der Ansicht mit den Datensatzdetails die Registerkarte Datenqualität aus, um die Ergebnisse anzuzeigen. Weitere Informationen zum Arbeiten mit Profilinformationen finden Sie unterErstellen und Arbeiten mit AWS Glue DataBrew Jobs profilieren.

Auf der Grundlage der Validierungsergebnisse handeln

Wenn ein DataBrew Profiljob abgeschlossen ist, wird ein CloudWatch Amazon-Event mit den Details dieses ausgeführten Jobs DataBrew gesendet. Wenn Sie Ihren Job auch für die Validierung von Datenqualitätsregeln konfiguriert haben, wird für jeden validierten Regelsatz ein Ereignis DataBrew gesendet. Das Ereignis enthält das Ergebnis (`SUCCEEDEDFAILED`, oder`ERROR`) und einen Link zum ausführlichen Bericht zur Datenqualitätsprüfung. Anschließend können Sie weitere Aktionen automatisieren, indem Sie je nach Status der Validierung die nächste Aktion aufrufen. Weitere Informationen zum Verbinden von Ereignissen mit Zielaktionen, wie Amazon SNS SNS-Benachrichtigungen,AWS Lambda Funktionsaufrufen und anderen, finden Sie unter Erste Schritte mit Amazon. EventBridge

Im Folgenden finden Sie ein Beispiel für ein Ereignis mit einem DataBrew Überprüfungsergebnis:

```
{
  "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"
  }
}
```

Sie können Attribute von Ereignissen wie `source` und verschachtelte Eigenschaften des `detail` Attributs verwenden`detail-type`, um Ereignismuster in Amazon Eventbridge zu erstellen. Ein Ereignismuster, das allen fehlgeschlagenen Validierungen eines DataBrew Jobs entspricht, würde beispielsweise so aussehen:

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

Ein Beispiel für die Erstellung eines Regelsatzes und die Validierung seiner Regeln finden Sie unter. Einen Regelsatz mit Datenqualitätsregeln erstellen Weitere Informationen zum Arbeiten mit CloudWatch Ereignissen in DataBrew finden Sie unter Automatisieren DataBrew mit Ereignissen CloudWatch

Einen Regelsatz mit Datenqualitätsregeln erstellen

Im folgenden Verfahren finden Sie ein Beispiel für die Erstellung eines Regelsatzes und dessen Anwendung auf einen Datensatz. Ein Regelsatz ist ein Regelsatz, der verschiedene Datenmetriken mit erwarteten Werten vergleicht. Anschließend können Sie diesen Regelsatz in einem Profiljob verwenden, um die darin enthaltenen Datenqualitätsregeln zu überprüfen.

Um einen Beispielregelsatz mit Datenqualitätsregeln zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole unter https://console.aws.amazon.com/databrew/.
2. Wählen Sie im Navigationsbereich DQ-REGELN und anschließend Datenqualitätsregelsatz erstellen aus.
3. Geben Sie einen Namen für Ihren Regelsatz ein. Geben Sie optional eine Beschreibung für Ihren Regelsatz ein.
4. Wählen Sie unter Zugeordneter Datensatz einen Datensatz aus, der dem Regelsatz zugeordnet werden soll.

   Nachdem Sie ein Dataset ausgewählt haben, können Sie das Dataset-Vorschaufenster auf der rechten Seite anzeigen.
5. Verwenden Sie die Vorschau im Bereich Dataset-Vorschau, um die Werte und das Schema für den Datensatz zu untersuchen, während Sie die zu erstellenden Datenqualitätsregeln festlegen. Die Vorschau kann Ihnen einen Einblick in mögliche Probleme geben, die Sie möglicherweise mit den Daten haben könnten.

   Einige Datenquellen, z. B. Datenbanken, unterstützen keine Datenvorschau. In diesem Fall können Sie einen Profiljob ausführen, ohne zuerst die Datenqualitätsregeln zu überprüfen. Anschließend können Sie mithilfe des Datenprofils Informationen zum Datenschema und zur Werteverteilung abrufen.
6. Schauen Sie auf der Registerkarte Empfehlungen nach, auf der einige Regelvorschläge aufgeführt sind, die Sie bei der Erstellung Ihres Regelsatzes verwenden können. Sie können alle, einige oder keine der Empfehlungen auswählen.

   Nachdem Sie die entsprechenden Empfehlungen ausgewählt haben, wählen Sie Zum Regelsatz hinzufügen aus.

   Dadurch werden Ihrem Regelsatz Regeln hinzugefügt. Überprüfen und ändern Sie die Parameter bei Bedarf. Beachten Sie, dass in Datenqualitätsregeln nur Spalten einfacher Typen wie Zeichenfolge, Zahlen und boolesche Werte verwendet werden können.
7. Wählen Sie Weitere Regel hinzufügen aus, um eine Regel hinzuzufügen, für die keine Empfehlungen gelten. Sie können die Regelnamen ändern, um die spätere Interpretation der Überprüfungsergebnisse zu erleichtern.
8. Verwenden Sie den Umfang der Datenqualitätsprüfung, um auszuwählen, ob bei jeder Prüfung in dieser Regel einzelne Spalten ausgewählt werden oder ob sie auf eine Gruppe von Spalten

angewendet werden sollen, die Sie auswählen. Wenn Ihr Datensatz beispielsweise mehrere numerische Spalten enthält, die Werte zwischen 0 und 100 haben sollten, können Sie die Regel einmal definieren und dann all diese Spalten auswählen, die nach dieser Regel geprüft werden sollen.

9. Wenn Ihre Regel mehr als eine Prüfung vorsieht, wählen Sie in der Dropdownliste Erfolgskriterien für Regeln aus, ob alle Prüfungen erfüllt werden sollen oder welche die Kriterien erfüllen.
10. Wählen Sie in der Dropdownliste Datenqualitätsprüfung eine Prüfung aus, die durchgeführt werden soll, um diese Regel zu verifizieren. Weitere Informationen zu verfügbaren Prüfungen finden Sie unterVerfügbare Schecks.
11. Wenn Sie für jede Spalte im Bereich der Datenqualitätsprüfung die Option Individuelle Prüfung ausgewählt haben, wählen Sie eine Spalte aus. Wählen Sie den Spaltennamen für diese Prüfung aus, oder geben Sie ihn ein.
12. Wählen Sie je nach Prüfung die Parameter aus. Einige Bedingungen akzeptieren nur bereitgestellte benutzerdefinierte Werte, andere unterstützen auch Verweise auf eine andere Spalte.
13. Wenn Sie „Prüfungen für Spaltenwerte“ auswählen, z. B. die Bedingung „Enthält“ für Zeichenkettenwerte, können Sie den Schwellenwert für das Bestehen angeben. Wenn Sie beispielsweise möchten, dass mindestens 95 Prozent der Werte die Bedingung erfüllen, müssen Sie als Bedingung für einen Schwellenwert die Option Größer als gleich wählen, 95 als Schwellenwert eingeben und „% (Prozent) Zeilen“ in der nächsten Dropdownliste im Bereich Schwellenwert belassen. Oder wenn Sie nicht mehr als 10 Zeilen möchten, in denen der Wert fehlt, wenn die Bedingung wahr ist, können Sie als Bedingung Weniger als gleich auswählen, 10 als Schwellenwert eingeben und Zeilen in der nächsten Dropdownliste auswählen. Bitte beachten Sie, dass Sie möglicherweise unterschiedliche Ergebnisse erhalten, wenn Sie bei der Validierung Stichproben unterschiedlicher Größe verwenden.
14. Fügen Sie bei Bedarf weitere Regeln hinzu.
15. Wählen Sie Regelsatz erstellen aus.

Einen Profiljob mithilfe eines Regelsatzes erstellen

Nachdem Sie wie oben beschrieben einen Regelsatz erstellt haben, werden Sie zur Seite mit den Datenqualitätsregeln weitergeleitet, auf der alle Regelsätze in Ihrem Konto angezeigt werden.

Um einen Profiljob mit einem Regelsatz zu erstellen

1. Wählen Sie den Namen des Regelsatzes, den Sie zuvor erstellt haben, um dessen Details anzuzeigen.
2. Wählen Sie Profiljob mit Regelsatz erstellen.

   Der Jobname wird automatisch ausgefüllt, Sie können ihn jedoch nach Bedarf ändern.
3. Für Job run sample können Sie wählen, ob der gesamte Datensatz oder eine begrenzte Anzahl von Zeilen ausgeführt werden soll.

   Wenn Sie sich dafür entscheiden, eine begrenzte Stichprobengröße durchzuführen, beachten Sie, dass sich die Ergebnisse bei bestimmten Regeln von denen des vollständigen Datensatzes unterscheiden können.
4. Wählen Sie unter Job-Output-Einstellungen einen S3-Speicherort für die Job-Ausgabe aus. Wählen Sie einen beliebigen Ordner in einem benannten Amazon S3 S3-Bucket aus, auf den Sie Zugriff haben. Wenn Sie einen Ordnernamen für diesen Bucket eingeben, der nicht existiert, wird dieser Ordner erstellt.

   Nach erfolgreichem Abschluss des Profiljobs enthält dieser Ordner Profile der Daten- und Datenqualitätsregelvalidierungsberichte im JSON-Format.
5. Beachten Sie, dass Ihr Regelsatz unter Datenqualitätsregeln unter Name des Datenqualitätsregelsatzes aufgeführt ist.
6. Wählen oder erstellen Sie unter Berechtigungen eine Rolle, um DataBrew Zugriff zum Lesen vom Amazon S3 S3-Eingabeort und zum Schreiben in den Jobausgabespeicherort zu gewähren. Wenn Sie noch keine Rolle bereit haben, wählen Sie Neue IAM-Rolle erstellen aus.
7. Ändern Sie bei Bedarf alle anderen optionalen Einstellungen wie unter beschrieben. Erstellen und Arbeiten mit AWS Glue DataBrew Jobs profilieren
8. Wählen Sie Job erstellen und ausführen.

Überprüfung der Validierungsergebnisse und Aktualisierung der Datenqualitätsregeln

Nach Abschluss Ihres Profiljobs können Sie die Überprüfungsergebnisse für Ihre Datenqualitätsregeln einsehen und Ihre Regeln bei Bedarf aktualisieren.

Um Validierungsdaten für Ihre Datenqualitätsregeln einzusehen

1. Wählen Sie in der DataBrew Konsole die Option Datenprofil anzeigen aus. Dadurch wird die Registerkarte Datenprofilübersicht für Ihren Datensatz angezeigt.
2. Wählen Sie die Registerkarte Datenqualitätsregeln. Auf dieser Registerkarte können Sie die Ergebnisse für alle Ihre Datenqualitätsregeln anzeigen.
3. Wählen Sie eine einzelne Regel aus, um weitere Informationen zu dieser Regel zu erhalten.

Für jede Regel, deren Überprüfung fehlgeschlagen ist, können Sie die erforderlichen Korrekturen vornehmen.

Um Ihre Datenqualitätsregeln zu aktualisieren

1. Wählen Sie im Navigationsbereich DQ RULES aus.
2. Wählen Sie unter Name des Regelsatzes für die Datenqualität den Datensatz aus, der die Regeln enthält, die Sie bearbeiten möchten.
3. Wählen Sie die Regel aus, die Sie ändern möchten, und klicken Sie dann auf Bearbeiten.
4. Nehmen Sie die erforderlichen Korrekturen vor und wählen Sie dann Regelsatz aktualisieren.
5. Führen Sie den Job erneut aus. Wiederholen Sie diesen Vorgang, bis alle Validierungen bestanden sind.

Verfügbare Schecks

In der folgenden Tabelle sind Verweise auf alle verfügbaren Bedingungen aufgeführt, die in Ihren Regeln verwendet werden können. Beachten Sie, dass aggregierte Bedingungen nicht mit nicht aggregierten Bedingungen in derselben Regel kombiniert werden können.

> **Note**
>
> Wenn SDK-Benutzer dieselbe Regel auf mehrere Spalten anwenden möchten, verwenden Sie das ColumnSelectorsAttribut einer Regel und geben Sie validierte Spalten entweder mit ihren Namen oder einem regulären Ausdruck an. In diesem Fall sollten Sie implizit CheckExpressionverwenden. Zum Beispiel, `"> :val"` um Werte in jeder der ausgewählten Spalten mit dem angegebenen Wert zu vergleichen. DataBrew verwendet implizite Syntax für die Definition FilterExpressionin dynamischen Datensätzen. Wenn Sie Spalte (n) für jede Prüfung einzeln angeben möchten, legen Sie das ColumnSelectorsAttribut nicht fest.

Geben Sie stattdessen einen expliziten Ausdruck an. Zum Beispiel “`:col > :val`” als CheckExpressionin einer Regel.

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax
Aggregieren Sie die Bedingungen für Datensätze	Anzahl der Zeilen		Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(ROWS_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
	Anzahl der Spalten		Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}`
	Doppelte Zeilen		Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT) < :val", "SubstitutionMap": {":val", "100"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax
				oder `"CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

Aggregieren Sie die Bedingungen für Spaltenstatistiken	Fehlende Werte		Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Doppelte Werte		Numerischer Vergleich mit benutzerdefinierten Werten	`"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Zulässige Werte		Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
				
				oder
				
				`"CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Eindeutige Werte		Numerischer Vergleich mit benutzerdefinierten Werten	`"CheckExpression": "AGG(DISTINCT_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "1000"}`
				
				oder
				
				`"CheckExpression": "AGG(DISTINCT_VALUES_PERCENTAGE) >= :val", "SubstitutionMap": {":val", "50"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Eindeutige Werte		Numerischer Vergleich mit benutzerdefinierten Werten	`"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Ausreißer	Z-score Schwellenwert	Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Statistik der Wertverteilung	Name der Statistik (siehe nächste Tabelle)	Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Mögliche `STAT_NAME` Werte finden Sie in der nächsten Tabelle

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Numerische Statistik	Name der Statistik (siehe nächste Tabelle)	Numerischer Vergleich mit benutzerdefiniertem Wert	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				oder
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Mögliche `STAT_NAME` Werte finden Sie in der nächsten Tabelle

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax
Nicht aggregiert (akzeptiert Schwellenwert)	Der Wert ist genau		Exakter Vergleich mit einer Werteliste	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	Der Wert ist nicht exakt		Der Wert sollte nicht exakt mit einem Wert aus einer Liste übereinstimmen	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Zeichenfolgenwerte		Vergleich einer Zeichenfolge mit einem benutzerdefinierten Wert oder einer anderen Zeichenfolgenspalte	`"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"}`
				
				oder
				
				`"CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Numerische Werte		Numerischer Vergleich mit einem benutzerdefinierten Wert oder einer anderen numerischen Spalte	`"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"}`
				
				oder
				
				`"CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}`

Bedingungstyp	Überprüfung der Datenqualität	Zusätzliche Parameter	Vergleichstyp	Beispiel für eine SDK-Syntax

	Länge der Wertzeich enfolge		Numerischer Vergleich mit einem benutzerd efinierten Wert oder einer anderen numerischen Spalte	`"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"}`
				
				oder
				
				`"CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}`

Numerische Vergleiche

DataBrew unterstützt die folgenden Operationen für numerische Vergleiche: Ist gleich (= =), Ist nicht gleich (! =), Kleiner als (<), Weniger als gleich (< =), Größer als (>), Größer als gleich (> =) und Liegt zwischen (is_between: val1 und:val2).

Vergleiche von Zeichenketten

Die folgenden Zeichenfolgenvergleiche werden unterstützt: Beginnt mit, Beginnt nicht mit, Endet mit, Endet nicht mit, Enthält, enthält nicht, Ist gleich, Ist nicht gleich, Stimmt überein, Stimmt nicht überein.

In der folgenden Tabelle werden verfügbare Statistiken angezeigt, die Sie für Wertverteilungsstatistiken und numerische Statistiken verwenden können:

Überprüfung der Datenqualität	Name der Statistik	Zusätzliche Parameter	SDK-Syntax
Statistiken zur Wertverteilung	Min		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	Max		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	Median		`"CheckExpression": "AGG(MEDIAN) >= :val", "SubstitutionMap": {":val", "50"}`

Überprüfung der Datenqualität	Name der Statistik	Zusätzliche Parameter	SDK-Syntax
	Mean		`"CheckExpression": "AGG(MEAN) <= :val", "SubstitutionMap": {":val", "10"}`
	Mode		`"CheckExpression": "AGG(MODE) > :val", "SubstitutionMap": {":val", "0"}`
	Standardabweichung		`"CheckExpression": "AGG(STANDARD_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Entropie		`"CheckExpression": "AGG(ENTROPY) > :val", "SubstitutionMap": {":val", "0"}`

Überprüfung der Datenqualität	Name der Statistik	Zusätzliche Parameter	SDK-Syntax
Numerische Statistik	Summe		`"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}`
	Kurtosis		`"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}`
	Schiefheit		`"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}`
	Varianz		`"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}`

Überprüfung der Datenqualität	Name der Statistik	Zusätzliche Parameter	SDK-Syntax
	Absolute Abweichung		`"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Quantil	Quantil: eins von '0,25', '0,5', '0,75'	`"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}`

Erstellen und Verwenden AWS Glue DataBrew projects

In AWS Glue DataBrew ist ein Projekt das Herzstück Ihrer Datenanalyse- und Transformationsbemühungen.

Wenn Sie ein Projekt erstellen, führen Sie zwei grundlegende Komponenten zusammen:

- Ein Datensatz, der nur Lesezugriff auf Ihre Quelldaten ermöglicht. Weitere Informationen finden Sie unter Verbindung zu Daten herstellen mit AWS Glue DataBrew.
- Ein Rezept, um DataBrew Datentransformationen auf den Datensatz anzuwenden. Weitere Informationen finden Sie unter Erstellen und Verwenden AWS Glue DataBrew recipes.

Die DataBrew Konsole präsentiert Ihr Projekt in einer hochgradig interaktiven, intuitiven Benutzeroberfläche. Sie ermutigt Sie, mit Hunderten von Datentransformationen zu experimentieren, um zu erfahren, wie sie funktionieren und welche Auswirkungen sie auf Ihre Daten haben.

Die Daten, die Sie in der Projektansicht sehen, sind ein Beispiel für Ihren Datensatz. Da Datensätze sehr umfangreich sein können und Tausende oder sogar Millionen von Zeilen umfassen können, trägt die Verwendung eines Beispiels dazu bei, dass die DataBrew Konsole reagiert, während Sie die Beispieldaten auf verschiedene Weise transformieren. Standardmäßig besteht die Stichprobe aus den ersten 500 Datenzeilen des Datensatzes. Sie können verschiedene Einstellungen für die Stichprobengröße und die ausgewählten Zeilen wählen.

DataBrew Hilft Ihnen bei der Transformation der Beispieldaten bei der Erstellung und Verfeinerung des Projektrezepts — eine schrittweise Abfolge der Transformationen, die Sie bisher angewendet haben. Ihr in Bearbeitung befindliches Rezept wird automatisch gespeichert, sodass Sie die Projektansicht jederzeit verlassen, später zurückkehren und dort weitermachen können, wo Sie aufgehört haben.

Wenn Ihr Rezept einsatzbereit ist, können Sie es veröffentlichen. Wenn Sie ein Rezept veröffentlichen, steht es dem DataBrew Job-Subsystem zur Verfügung, wo Sie das Rezept auf Ihren gesamten Datensatz anwenden oder ein umfangreiches Datenprofil erstellen können, das Ihnen einen Überblick über die Struktur, den Inhalt und die statistischen Merkmale Ihrer Daten gibt.

Themen

- Erstellen eines Projekts
- Überblick über eine DataBrew Projektsitzung

- Löschen eines Projekts

Erstellen eines Projekts

Gehen Sie wie folgt vor, um ein Projekt zu erstellen.

So erstellen Sie ein Projekt

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole.
2. Wählen Sie im Navigationsbereich PROJEKTE aus. Wählen Sie dann Projekt erstellen aus.
3. Geben Sie einen Namen für Ihr Projekt ein. Wählen Sie dann ein Rezept aus, das Sie an Ihr Projekt anhängen möchten:

   - Wählen Sie Neues Rezept erstellen, wenn Sie von vorne beginnen. Dadurch wird ein neues, leeres Rezept erstellt und an Ihr Projekt angehängt.
   - Wählen Sie Bestehendes Rezept bearbeiten, wenn Sie ein zuvor veröffentlichtes Rezept haben, das Sie für dieses Projekt verwenden möchten. Wenn das Rezept derzeit an ein anderes Projekt angehängt ist oder Jobs dafür definiert sind, können Sie es nicht in Ihrem neuen Projekt verwenden. Wählen Sie Rezepte durchsuchen, um zu sehen, welche Rezepte verfügbar sind.
   - Wählen Sie „Schritte aus Rezept importieren“, wenn Sie bereits über ein bereits veröffentlichtes Rezept verfügen und dessen Schritte importieren möchten, und gehen Sie dann wie folgt vor:
     1. Wähle „Rezepte durchsuchen“, um zu sehen, welche Rezepte verfügbar sind.
     2. Wählen Sie die veröffentlichte Version des Rezepts aus, das Sie verwenden möchten. Ein Rezept kann mehrere Versionen haben, je nachdem, wie oft Sie es während der Arbeit in der Projektansicht veröffentlicht haben.
     3. Wählen Sie Rezeptschritte anzeigen aus, um die Datentransformationen im Rezept zu untersuchen.
4. Nachdem Sie ein Rezept erstellt haben, wählen Sie im Bereich Datensatz auswählen den Datensatz aus, mit dem Sie arbeiten möchten:

   - Meine Datensätze — Wählen Sie einen Datensatz aus, den Sie zuvor erstellt haben. Weitere Informationen finden Sie unter Erstellen eines Projekts.)
   - Beispieldateien — Erstellen Sie einen neuen Datensatz auf der Grundlage von Beispieldaten, die von AWS verwaltet werden. Diese Beispieldaten eignen sich hervorragend, um

herauszufinden, was alles DataBrew möglich ist, ohne eigene Daten angeben zu müssen. Stellen Sie sicher, dass Sie einen Namen für Ihren Datensatz eingeben.

- Neuer Datensatz — Erstellen Sie einen neuen Datensatz. Weitere Informationen finden Sie unter Erstellen eines Projekts.

5. Wählen Sie für Zugriffsberechtigungen eine AWS Identity and Access Management(IAM-) Rolle aus, die das Lesen von Ihrem Amazon S3 S3-Eingabespeicherort ermöglicht DataBrew . Für einen S3-Standort, der Ihrem AWS Konto gehört, können Sie die vom `AwsGlueDataBrewDataAccessRole` Service verwaltete Rolle wählen. Auf diese Weise können Sie DataBrew auf S3-Ressourcen zugreifen, die Sie besitzen.
6. Im Bereich Sampling finden Sie Optionen, mit denen DataBrew Sie eine Stichprobe von Daten aus Ihrem Datensatz erstellen können.

   Wählen Sie unter Typ aus, wie Zeilen aus Ihrem Datensatz abgerufen werden DataBrew sollen:

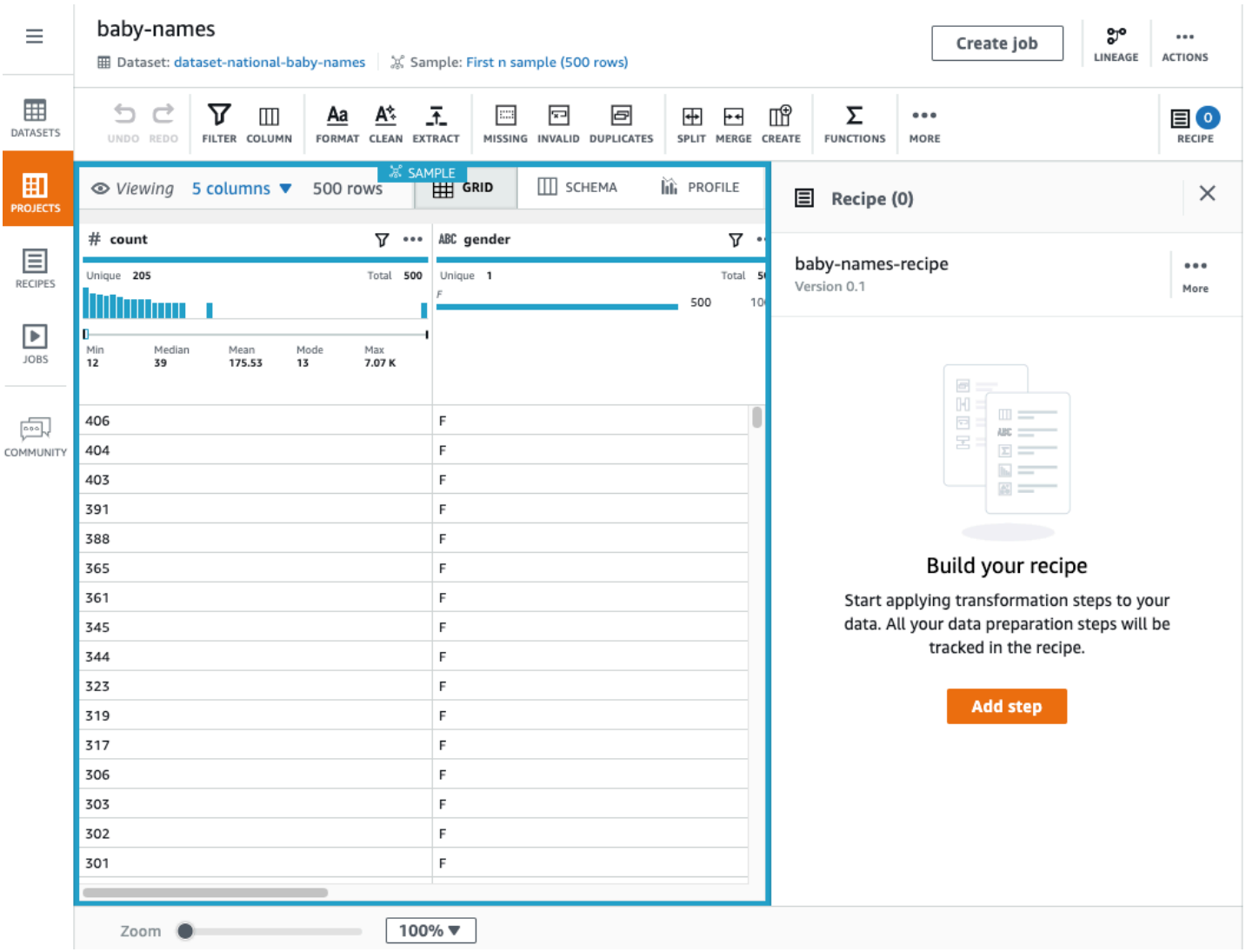
   - Verwenden Sie First n rows, um eine Stichprobe zu erstellen, die auf den ersten Zeilen im Datensatz basiert.
   - Verwenden Sie Zufällige Zeilen, um eine Stichprobe zu erstellen, die auf einer zufälligen Auswahl von Zeilen im Datensatz basiert.
   - Wählen Sie die Anzahl der Zeilen, die in der Stichprobe erscheinen sollen: 500, 1.000, 2.500 oder eine benutzerdefinierte Stichprobengröße, bis zu einem Maximum von 5.000 Zeilen. Ein kleinerer Stichprobenumfang DataBrew ermöglicht eine schnellere Durchführung von Transformationen, wodurch Sie Zeit bei der Entwicklung Ihres Rezepts sparen. Ein größerer Stichprobenumfang spiegelt die Zusammensetzung der zugrunde liegenden Quelldaten genauer wider. Die Initialisierung von Projektsitzungen und interaktive Transformationen sind jedoch langsamer.
7. (Optional) Wählen Sie „Tags“, um Ihrem Datensatz Tags hinzuzufügen.

   Tags sind einfache Beschriftungen, die aus einem benutzerdefinierten Schlüssel und einem optionalen Wert bestehen und die das Verwalten, Suchen und Filtern von DataBrew Projekten nach Zweck, Eigentümer, Umgebung oder anderen Kriterien erleichtern können.
8. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Job erstellen.

DataBrew erstellt bei Bedarf einen neuen Datensatz, erstellt bei Bedarf ein neues Rezept, erstellt das Datenbeispiel und erstellt eine interaktive Projektsitzung. Dieser Vorgang kann einige Minuten in Anspruch nehmen. Wenn das Projekt einsatzbereit ist, können Sie mit der Arbeit an der Datenprobe beginnen.

Überblick über eine DataBrew Projektsitzung

In einer DataBrew Projektsitzung arbeiten Sie in einem interaktiven Arbeitsbereich.



Im linken Bereich wird die aktuelle Ansicht Ihrer Daten angezeigt. Im rechten Bereich wird das Transformationsrezept des Projekts angezeigt, das derzeit leer ist.

In der oberen rechten Ecke des Datenrasters befinden sich drei Registerkarten: `GRID`, `SCHEMA` und. `PROFILE` Wenn Sie eine dieser Registerkarten auswählen, wird eine entsprechende Ansicht im Arbeitsbereich angezeigt. Diese Ansichten werden im Folgenden beschrieben.

Rasteransicht

Die Rasteransicht ist die Standardansicht, in der das Beispiel im Tabellenformat angezeigt wird. Gehen Sie wie folgt vor, um einen kurzen Überblick über die Rasteransicht zu erhalten.

Um eine exemplarische Vorgehensweise durch die Rasteransicht zu machen

1. Sehen Sie sich zunächst den gesamten Bereich an:

   a. Scrollen Sie nach links und rechts, um alle Spalten zu sehen.

   b. Scrollen Sie nach oben und unten, um alle Datenwerte zu sehen.

   c. Verwenden Sie das Zoom-Steuerelement am unteren Rand des Arbeitsbereichs, um den Vergrößerungsgrad des Rasters anzupassen.

2. In der oberen rechten Ecke sehen Sie, wie viele Spalten der Stichprobe angezeigt werden und wie viele Zeilen die Stichprobe aktuell enthält.

   Um zu ändern, welche Spalten angezeigt werden, wählen Sie den Link N Spalten (wobei N die Anzahl der aktuell angezeigten Spalten ist). Wählen Sie die gewünschten Spalten aus und wählen Sie Ausgewählte Spalten anzeigen aus.

3. Jetzt können Sie anfangen, mit DataBrew Transformationen zu experimentieren. Gehen Sie wie folgt vor:

   a. Wählen Sie in der Transformationswerkzeugleiste „Format wählen“, „In Großbuchstaben ändern“.

   b. Wählen Sie für Quellspalte eine Spalte aus, die Zeichendaten enthält.

   c. Übernehmen Sie für die anderen Einstellungen die Standardwerte.

   d. Um zu sehen, wie die transformierten Daten aussehen werden, wählen Sie „Änderungen in der Vorschau anzeigen“. Um diese Transformation dann zu Ihrem Rezept hinzuzufügen, wählen Sie Anwenden.

   Immer wenn Sie eine Datentransformation anwenden, DataBrew fügt sie der Arbeitskopie Ihres Rezepts hinzu. Dies wird auf der rechten Seite Ihres Arbeitsbereichs angezeigt.

4. Gehen Sie wie folgt vor:

   a. Wählen Sie in der Transformationswerkzeugleiste „Erstellen“, „Basierend auf einer Funktion“.

   b. Wählen Sie für Funktion auswählen die Option`SQUARE ROOT`.

   c. Wählen Sie für Quellspalte eine Spalte aus, die numerische Daten enthält.

   d. Belassen Sie die anderen Einstellungen auf ihren Standardeinstellungen,.

e. Wählen Sie „Änderungen in der Vorschau anzeigen“, um zu sehen, wie die transformierten Daten aussehen. Um diese Transformation dann zu Ihrem Rezept hinzuzufügen, wählen Sie Anwenden.

5. Reduzieren Sie den Rezeptbereich oben rechts, indem Sie REZEPT wählen. Um den Rezeptbereich zu erweitern, wählen Sie erneut RECIPE.

Veröffentlichen Sie eine neue Version Ihres Rezepts

Wenn Sie weitere Transformationen anwenden, nimmt die Anzahl der Schritte im Rezept zu. Sie können jederzeit eine neue Version Ihres Rezepts veröffentlichen. Wenn Sie ein Rezept veröffentlichen, ist es an anderer Stelle verfügbar DataBrew. Auf diese Weise können Sie einen Rezeptjob ausführen, um Ihren gesamten Datensatz zu transformieren, anstatt nur das Projektdatenbeispiel zu transformieren.

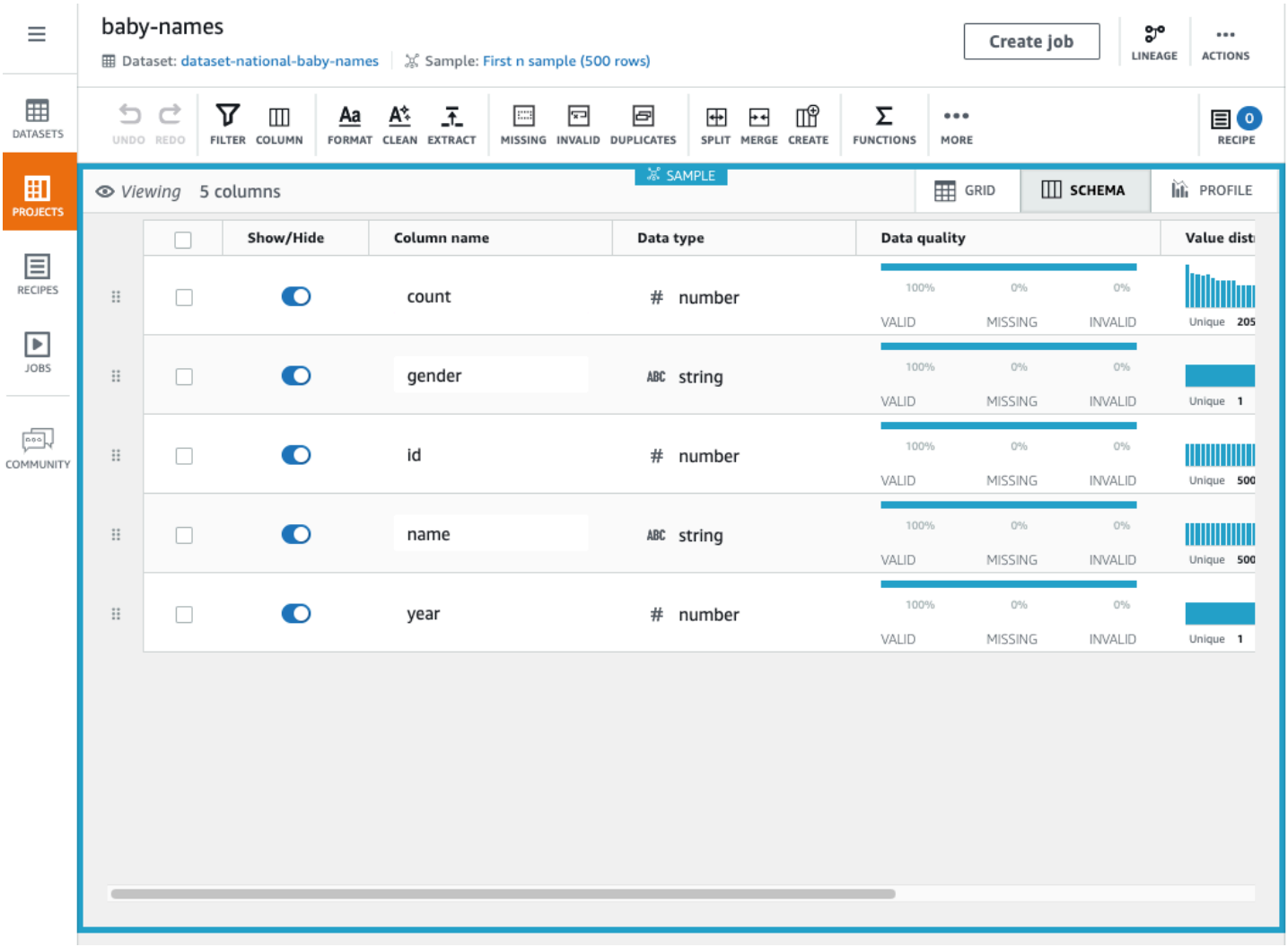
Das Veröffentlichen von Rezepten fördert auch einen schrittweisen, iterativen Ansatz bei der Rezeptentwicklung: Sie können nach und nach neue Versionen Ihres Rezepts veröffentlichen, sodass Sie bei Bedarf auf eine Rezeptversion zurückgreifen können, die zuletzt als funktionierend bekannt war.

Um eine neue Version eines Rezepts zu veröffentlichen

- Wählen Sie im Rezeptbereich die Option Veröffentlichen aus. Geben Sie eine Beschreibung für diese Version des Rezepts ein und wählen Sie „Veröffentlichen“.

Schema-Ansicht

Wenn Sie die Registerkarte SCHEMA wählen, ändert sich die Ansicht, wie im folgenden Screenshot gezeigt.



In der Schemaansicht können Sie Statistiken zu den Datenwerten in jeder Spalte sehen.

Wählen Sie in der Spalte ganz links neben Show/Hideeine der Datenspalten aus. Der Bereich mit den Spaltendetails wird auf der rechten Seite angezeigt. In diesem Bereich wird eine Zusammenfassung der Statistiken für die Spaltenwerte angezeigt.

Sie können eine Spalte umbenennen, indem Sie einen neuen Namen für den Spaltennamen eingeben.

Sie können die Reihenfolge der Spalten ändern, indem Sie die Spalten per Drag-and-Drop verschieben.

Profilansicht

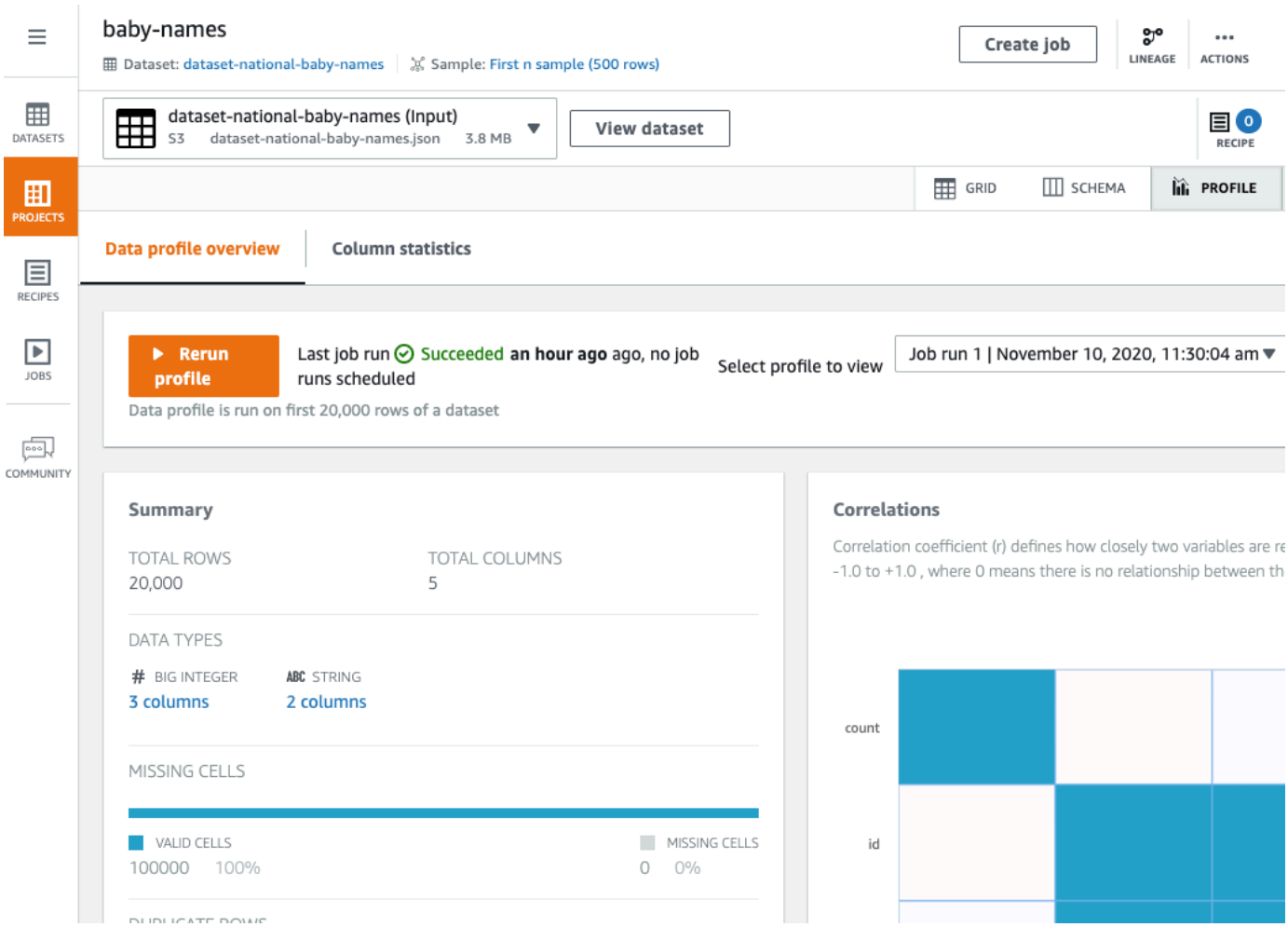
Wenn Sie die Registerkarte PROFIL wählen, können Sie detaillierte volumetrische Informationen zu Ihrem Projekt einsehen. Bevor Sie dies tun, führen Sie einen DataBrew Job aus, um das Profil zu erstellen.

Um eine exemplarische Vorgehensweise durch die Profilansicht zu erhalten

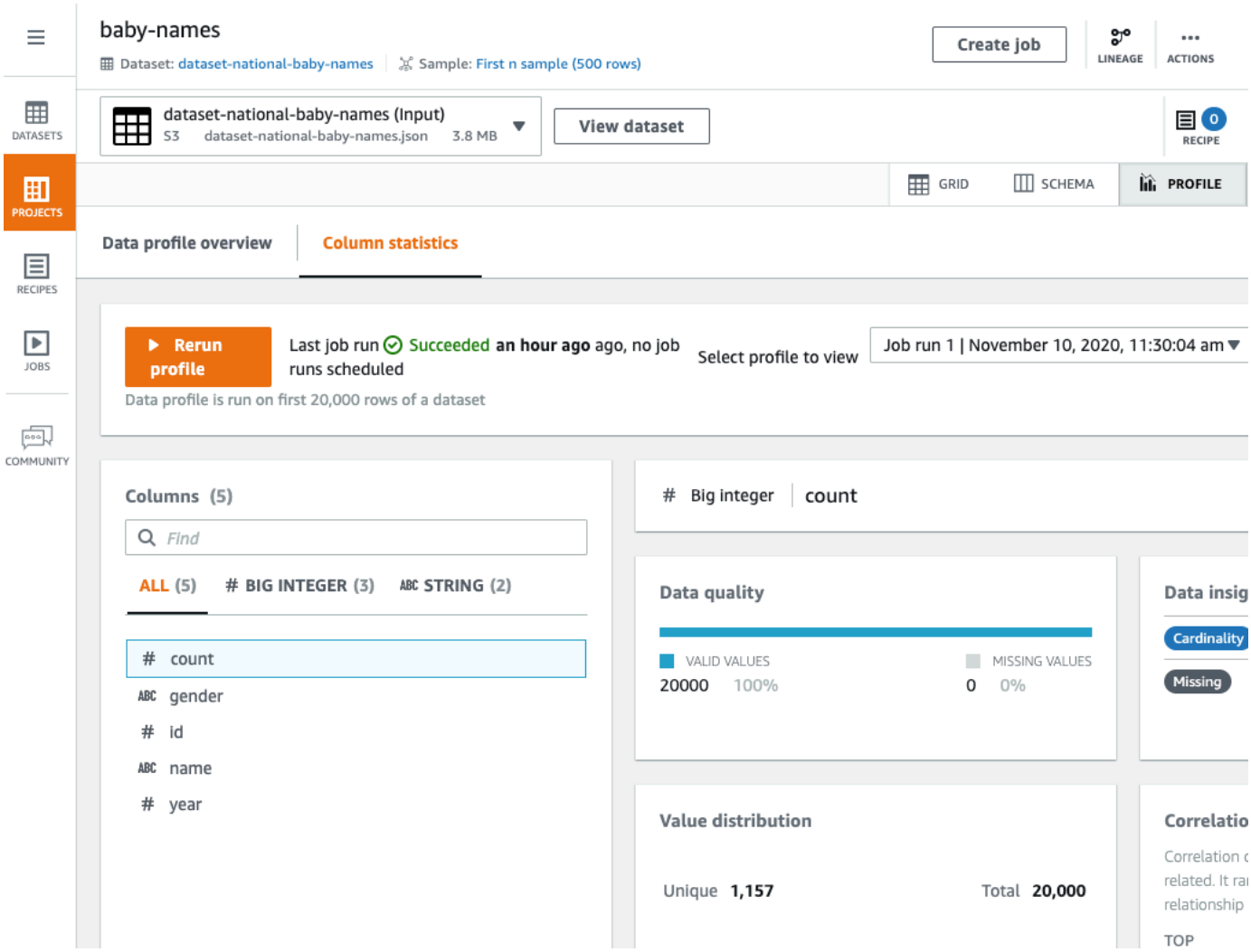
1. Wählen Sie Job erstellen und geben Sie einen Namen für Ihren Job ein.
2. Wählen Sie für Job-Ausgabe CSV als Dateityp aus.
3. Suchen oder erstellen Sie einen Amazon S3 S3-Bucket und -Ordner in Ihrem AWS Konto, aus dem die Job-Ausgabe geschrieben werden DataBrew soll:
   - Wenn Sie diesen Amazon S3 S3-Bucket und diesen Ordner bereits haben, wählen Sie Durchsuchen und suchen Sie sie. Stellen Sie sicher, dass Sie für beide Schreibberechtigungen haben.
   - Wenn Sie diesen Amazon S3 S3-Bucket und diesen Ordner nicht haben, erstellen Sie sie:
     1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
     2. Wenn Sie keinen Amazon S3 S3-Bucket haben, wählen Sie Create Bucket. Geben Sie unter Bucket-Name einen eindeutigen Namen für Ihren neuen Bucket ein. Wählen Sie Create Bucket (Bucket erstellen) aus.
     3. Wählen Sie aus der Liste der Buckets den aus, den Sie verwenden möchten.
     4. Wählen Sie Create folder. Geben Sie `databrew-output` als Ordnername den Namen Ordner erstellen ein und wählen Sie ihn aus.
4. Wählen Sie für Zugriffsberechtigungen eine IAM-Rolle aus, die das Schreiben in Ihren Amazon S3 S3-Ausgabespeicherort ermöglicht DataBrew .

   Für einen S3-Standort, der Ihrem AWS Konto gehört, können Sie die vom `AwsGlueDataBrewDataAccessRole` Service verwaltete Rolle wählen. Auf diese Weise können Sie DataBrew auf S3-Ressourcen zugreifen, die Sie besitzen.
5. Behalten Sie die Standardeinstellungen für die anderen Einstellungen bei und wählen Sie Job erstellen und ausführen.
6. Nachdem der Job vollständig ausgeführt wurde, zeigt der Workspace eine grafische Zusammenfassung des Datenprofils an.

Auf der Registerkarte „Datenprofilübersicht“ wird eine allgemeine Zusammenfassung der Eigenschaften Ihrer Daten angezeigt, wie in der folgenden Abbildung dargestellt.



Die Registerkarte Spaltenstatistiken zeigt eine spaltenweise Aufschlüsselung der Datenwerte:



Löschen eines Projekts

Wenn Sie ein Projekt nicht mehr benötigen, können Sie es löschen.

So löschen Sie ein Projekt

1. Wählen Sie im Navigationsbereich PROJEKTE aus.
2. Wählen Sie das Projekt aus, das Sie löschen möchten, und wählen Sie dann für Aktionen die Option Löschen aus. .

Erstellen und Verwenden AWS Glue DataBrew recipes

In DataBrew: Ein Rezept besteht aus einer Reihe von Schritten zur Datentransformation. Sie können diese Schritte auf eine Stichprobe Ihrer Daten anwenden oder dasselbe Rezept auf einen Datensatz anwenden.

Der einfachste Weg, ein Rezept zu entwickeln, besteht darin, ein DataBrew Projekt zu erstellen, in dem Sie interaktiv mit einer Stichprobe Ihrer Daten arbeiten können. Weitere Informationen finden Sie unter. Erstellen und Verwenden AWS Glue DataBrew projects Im Rahmen des Workflows zur Projekterstellung wird ein neues (leeres) Rezept erstellt und an das Projekt angehängt. Anschließend können Sie mit der Erstellung Ihres Rezepts beginnen, indem Sie Datentransformationen hinzufügen.

> **Note**
>
> Sie können bis zu 100 Datentransformationen in ein einziges DataBrew Rezept aufnehmen.

Während Sie mit der Entwicklung Ihres Rezepts fortfahren, können Sie Ihre Arbeit speichern, indem Sie das Rezept veröffentlichen. DataBrew verwaltet eine Liste der veröffentlichten Versionen für Ihr Rezept. Sie können jede veröffentlichte Version in einem Rezept-Job verwenden, um das Rezept (in einem Rezept-Job) auszuführen und Ihren Datensatz zu transformieren. Sie können auch eine Kopie der Rezeptschritte herunterladen, sodass Sie das Rezept in anderen Projekten oder anderen Datensatztransformationen wiederverwenden können.

Sie können DataBrew Rezepte auch programmgesteuert entwickeln, indem Sie das AWS Command Line Interface(AWS CLI) oder eines der SDKs verwenden.AWSIn der DataBrew API werden Transformationen als Rezeptaktionen bezeichnet.

> **Note**
>
> In einer interaktiven DataBrew Projektsitzung führt jede Datentransformation, die Sie anwenden, zu einem Aufruf der DataBrew API. Diese API-Aufrufe erfolgen automatisch, ohne dass Sie die Details hinter den Kulissen kennen müssen.

Auch wenn Sie kein Programmierer sind, ist es hilfreich, die Struktur eines Rezepts und die DataBrew Organisation der Rezeptaktionen zu verstehen.

Themen

- Veröffentlichung einer neuen Rezeptversion
- Definition einer Rezeptstruktur

Veröffentlichung einer neuen Rezeptversion

Sie veröffentlichen neue Versionen eines Rezepts in einer interaktiven DataBrew Projektsitzung.

Um eine neue Rezeptversion zu veröffentlichen

1. Wählen Sie im Rezeptbereich die Option Veröffentlichen aus.
2. Geben Sie eine Beschreibung für diese Version des Rezepts ein und wählen Sie „Veröffentlichen“.

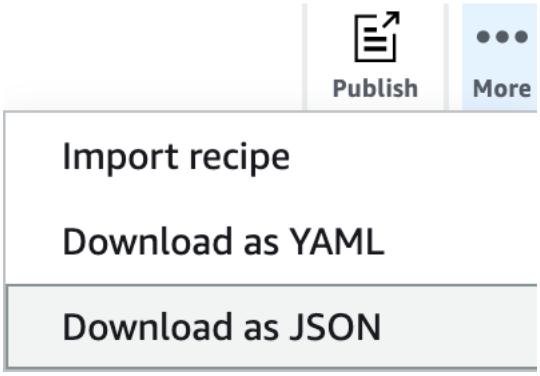
Sie können alle Ihre veröffentlichten Rezepte und deren Versionen anzeigen, indem Sie im Navigationsbereich PROJEKTE auswählen.

Definition einer Rezeptstruktur

Wenn Sie zum ersten Mal ein Projekt mit der DataBrew Konsole erstellen, definieren Sie ein Rezept, das diesem Projekt zugeordnet werden soll. Wenn Sie noch kein Rezept haben, erstellt die Konsole eines für Sie.

Während Sie in der Konsole mit Ihrem Projekt arbeiten, verwenden Sie die Transformationssymbolleiste, um Aktionen auf die Beispieldaten aus Ihrem Datensatz anzuwenden. In der Konsole werden die Rezeptschritte und die Reihenfolge dieser Schritte angezeigt, während Sie mit der Rezepturerstellung fortfahren. Sie können das Rezept wiederholen und verfeinern, bis Sie mit den Schritten zufrieden sind.

In Erste Schritte mit AWS Glue DataBrew erstellen Sie ein Rezept zur Transformation eines Datensatzes berühmter Schachpartien. Sie können eine Kopie der Rezeptschritte herunterladen, indem Sie Als JSON herunterladen oder Als YAML herunterladen wählen, wie im folgenden Screenshot gezeigt.



Die heruntergeladene JSON-Datei enthält Rezeptaktionen, die den Transformationen entsprechen, die Sie Ihrem Rezept hinzugefügt haben.

Ein neues Rezept hat keine Schritte. Sie können ein neues Rezept als leere JSON-Liste darstellen, wie im Folgenden gezeigt.

```
[ ]
```

Es folgt ein Beispiel für eine solche Datei, für`chess-project-recipe`. Die JSON-Liste enthält mehrere Objekte, die die Rezeptschritte beschreiben. Jedes Objekt in der JSON-Liste ist in geschweifte Klammern () `{ }` eingeschlossen. Die JSON-Zeilen sind durch Kommas getrennt.

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

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

```
                "sourceColumn": "victory_status",
                "value": "checkmate"
            }
        }
    },
    {
        "Action": {
            "Operation": "REPLACE_TEXT",
            "Parameters": {
                "pattern": "resign",
                "sourceColumn": "victory_status",
                "value": "other player resigned"
            }
        }
    },
    {
        "Action": {
            "Operation": "REPLACE_TEXT",
            "Parameters": {
                "pattern": "outoftime",
                "sourceColumn": "victory_status",
                "value": "ran out of time"
            }
        }
    }
]
```

Es ist einfacher zu erkennen, dass jede Aktion eine einzelne Zeile ist, wenn wir nur neue Zeilen für neue Aktionen hinzufügen, wie im Folgenden gezeigt.

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

Die Aktionen werden nacheinander in derselben Reihenfolge wie in der Datei ausgeführt:

- `REMOVE_VALUES`— Um alle Spiele herauszufiltern, bei denen die Bewertung eines Spielers unter 1.800 liegt, ist die Mindestwertung erforderlich, um ein Schachspieler der Klasse A zu sein. Es gibt zwei Fälle dieser Aktion: einmal, um Spieler auf der schwarzen Seite zu entfernen, die nicht mindestens Spieler der Klasse A sind, und ein anderes, um Spieler auf der weißen Seite zu entfernen, die nicht auf diesem Level sind.
- `GROUP_BY`— Um die Daten zusammenzufassen. In diesem Fall sortiert GROUP_BY die Zeilen anhand der Werte von `winner` (`black`und) in Gruppen. `white` Jede dieser Gruppen wird dann weiter unterteilt, wobei die Zeilen anhand der Werte von `victory_status` (, `mate` `resign`und`outoftime`) in Untergruppen sortiert werden. `draw` Schließlich wird die Anzahl der Vorkommen für jede Untergruppe gezählt. Die daraus resultierende Zusammenfassung ersetzt dann die ursprüngliche Datenstichprobe.
- `REMOVE_VALUES`— Um die Ergebnisse von Spielen zu löschen, die mit endeten`draw`.
- `REPLACE_TEXT`— Um die Werte für zu ändern`victory_status`. Es gibt drei Vorkommen dieser Aktion — jeweils eines für`mate`, und`resign`. `oufoftime`

In einer interaktiven DataBrew Projektsitzung `RecipeAction` entspricht jede Sitzung einer Datentransformation, die Sie auf eine Datenprobe anwenden.

DataBrew bietet über 200 Rezeptaktionen. Weitere Informationen finden Sie unter Rezeptschritt und Funktionsreferenz.

Verwenden von Bedingungen

Sie können Bedingungen verwenden, um den Umfang einer Rezeptaktion einzuschränken. Bedingungen werden bei Transformationen verwendet, bei denen Daten gefiltert werden, z. B. beim Entfernen unerwünschter Zeilen auf der Grundlage eines bestimmten Spaltenwerts.

Schauen wir uns ein Rezept mit Aktionen von genauer an. `chess-project-recipe`

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

Diese Transformation liest die Werte in der `black_rating` Spalte. Die `ConditionExpressions` Liste bestimmt die Filterkriterien: Jede Zeile mit einem `black_rating` Wert von weniger als 1.800 wird aus dem Datensatz entfernt.

Eine nachfolgende Transformation im Rezept bewirkt dasselbe, für`white_rating`. Auf diese Weise sind die Daten auf Spiele beschränkt, bei denen jeder Spieler (schwarz oder weiß) mit Klasse A oder höher bewertet wird.

Hier ist ein weiteres Beispiel für eine Bedingung, die auf eine Spalte mit Charakterdaten angewendet wird.

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

```
    ]
  }
```

Diese Transformation liest die Werte in der `winner` Spalte, sucht nach dem Wert `draw` und entfernt diese Zeilen. Auf diese Weise sind die Daten nur auf die Spiele beschränkt, bei denen es einen klaren Gewinner gab.

DataBrew unterstützt die folgenden Bedingungen:

- `IS`— Der Wert in der Spalte entspricht dem Wert, der in der Bedingung angegeben wurde.
- `IS_NOT`— Der Wert in der Spalte entspricht nicht dem Wert, der in der Bedingung angegeben wurde.
- `IS_BETWEEN`— Der Wert in der Spalte liegt zwischen den `LESS_THAN_EQUAL` Parametern `GREATER_THAN_EQUAL` und.
- `CONTAINS`— Der Zeichenkettenwert in der Spalte enthält den Wert, der in der Bedingung angegeben wurde.
- `NOT_CONTAINS`— Der Wert in der Spalte enthält nicht die Zeichenfolge, die in der Bedingung angegeben wurde.
- `STARTS_WITH`— Der Wert in der Spalte beginnt mit der Zeichenfolge, die in der Bedingung angegeben wurde.
- `NOT_STARTS_WITH`— Der Wert in der Spalte beginnt nicht mit der Zeichenfolge, die in der Bedingung angegeben wurde.
- `ENDS_WITH`— Der Wert in der Spalte endet mit der Zeichenfolge, die in der Bedingung angegeben wurde.
- `NOT_ENDS_WITH`— Der Wert in der Spalte endet nicht mit der Zeichenfolge, die in der Bedingung angegeben wurde.
- `LESS_THAN`— Der Wert in der Spalte ist kleiner als der Wert, der in der Bedingung angegeben wurde.
- `LESS_THAN_EQUAL`— Der Wert in der Spalte ist kleiner oder gleich dem Wert, der in der Bedingung angegeben wurde.
- `GREATER_THAN`— Der Wert in der Spalte ist größer als der Wert, der in der Bedingung angegeben wurde.
- `GREATER_THAN_EQUAL`— Der Wert in der Spalte ist größer oder gleich dem Wert, der in der Bedingung angegeben wurde.
- `IS_INVALID`— Der Wert in der Spalte hat einen falschen Datentyp.

- `IS_MISSING`— Die Spalte enthält keinen Wert.

Erstellen, Ausführen und Planen AWS Glue DataBrew jobs

AWS Glue DataBrew hat ein Job-Subsystem, das zwei Zwecken dient:

1. Anwenden eines Rezepts für die Datentransformation auf einen DataBrew Datensatz. Sie tun dies mit einem DataBrew Rezeptjob.
2. Analysieren eines Datensatzes, um ein umfassendes Profil der Daten zu erstellen. Sie tun dies mit einem DataBrew Profiljob.

Themen

- Erstellen und Arbeiten mit AWS Glue DataBrew Rezeptjobs
- Erstellen und Arbeiten mit AWS Glue DataBrew Jobs profilieren

Erstellen und Arbeiten mit AWS Glue DataBrew Rezeptjobs

Verwenden Sie einen DataBrew Rezeptjob, um die Daten in einem DataBrew Datensatz zu bereinigen und zu normalisieren und das Ergebnis an einen Ausgabeort Ihrer Wahl zu schreiben. Die Ausführung eines Rezeptjobs hat keine Auswirkungen auf den Datensatz oder die zugrunde liegenden Quelldaten. Wenn ein Job ausgeführt wird, stellt er schreibgeschützt eine Verbindung zu den Quelldaten her. Die Jobausgabe wird in einen Ausgabespeicherort geschrieben, den Sie in Amazon S3 AWS Glue Data Catalog, der oder einer unterstützten JDBC-Datenbank definieren.

Gehen Sie wie folgt vor, um einen DataBrew Rezeptjob zu erstellen.

Um einen Rezeptjob zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole unter https://console.aws.amazon.com/databrew/.
2. Wählen Sie im Navigationsbereich die Option JOBS aus, wählen Sie die Registerkarte Rezepturaufträge und dann Job erstellen aus.
3. Geben Sie einen Namen für Ihren Job ein und wählen Sie dann Create a recipe job aus.
4. Geben Sie unter Job-Eingabe Details zu dem Job ein, den Sie erstellen möchten: den Namen des zu verarbeitenden Datensatzes und das zu verwendende Rezept.

   Ein Rezeptjob verwendet ein DataBrew Rezept, um einen Datensatz zu transformieren. Um ein Rezept zu verwenden, stellen Sie sicher, dass Sie es zuerst veröffentlichen.

5. Konfigurieren Sie Ihre Einstellungen für die Jobausgabe.

   Geben Sie ein Ziel für Ihre Jobausgabe an. Wenn Sie keine DataBrew Verbindung für Ihr Ausgabeziel konfiguriert haben, konfigurieren Sie diese zunächst auf der Registerkarte DATENSÄTZE, wie unter beschriebenUnterstützte Verbindungen für Datenquellen und Ausgaben. Wählen Sie eines der folgenden Ausgabeziele:

   - Amazon S3, mit oder ohne AWS Glue Data Catalog Unterstützung
   - Amazon Redshift, mit oder ohne Unterstützung AWS Glue Data Catalog
   - JDBC
   - Snowflake-Tabellen
   - Amazon RDS-Datenbanktabellen mit AWS Glue Data Catalog Unterstützung. Amazon RDS-Datenbanktabellen unterstützen die folgenden Datenbank-Engines:
     - Amazon Aurora
     - MySQL
     - Oracle
     - PostgreSQL
     - Microsoft SQL Server
   - Amazon S3 mit AWS Glue Data Catalog Unterstützung.

   DataBrew Unterstützt bei der AWS Glue Data CatalogAWS Lake Formation Ausgabe nur das Ersetzen vorhandener Dateien. Bei diesem Ansatz werden die Dateien ersetzt, um Ihre bestehenden Lake Formation Formation-Berechtigungen für Ihre Datenzugriffsrolle beizubehalten. Außerdem hat der Amazon S3 S3-Standort aus der DataBrew AWS Glue Data Catalog Tabelle Vorrang. Daher können Sie den Amazon S3 S3-Speicherort nicht überschreiben, wenn Sie einen Rezeptjob erstellen.

   In einigen Fällen unterscheidet sich der Amazon S3 S3-Speicherort in der Jobausgabe vom Amazon S3 S3-Speicherort in der Datenkatalogtabelle. In diesen Fällen DataBrew aktualisiert die Auftragsdefinition automatisch mit dem Amazon S3 S3-Standort aus der Katalogtabelle. Dies geschieht, wenn Sie Ihre vorhandenen Jobs aktualisieren oder starten.

6. Nur für Amazon S3 S3-Ausgabeziele haben Sie weitere Auswahlmöglichkeiten:

   a. Wählen Sie eines der verfügbaren Datenausgabeformate für Amazon S3, optionale Komprimierung und ein optionales benutzerdefiniertes Trennzeichen. Die unterstützten

Trennzeichen für Ausgabedateien sind dieselben wie für die Eingabe: Komma, Doppelpunkt, Semikolon, senkrechter Strich, Tabulator, Caret, umgekehrter Schrägstrich und Leerzeichen. Einzelheiten zur Formatierung finden Sie in der folgenden Tabelle.

Format	Dateierweiterung (unkomprimiert)	Dateierweiterungen (komprimiert)
Comma-separated Werte	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated Werte	`.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 Parkett	Nicht unterstützt	`.glue.parquet.snappy`
Apache Avro	`.avro`	`.avro.snappy` , `.avro.gz`, `.avro.lz4` , `.avro.bz2` , `.avro.deflate` , `.avro.br`
Apache ORC	`.orc`	`.orc.snappy` , `.orc.lzo`, `.orc.zlib`
XML	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`

Format	Dateierweiterung (unkomprimiert)	Dateierweiterungen (komprimiert)
JSON (nur JSON-Zeilenformat)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Hyper Tableau	Nicht unterstützt	Nicht zutreffend

b. Wählen Sie aus, ob eine einzelne Datei oder mehrere Dateien ausgegeben werden sollen. Es gibt drei Optionen für die Dateiausgabe mit Amazon S3:

- Dateien automatisch generieren (empfohlen) — Hat die optimale Anzahl von Ausgabedateien DataBrew ermittelt.
- Einzeldateiausgabe — Bewirkt, dass eine einzelne Ausgabedatei generiert wird. Diese Option kann zu zusätzlicher Job-Ausführungszeit führen, da eine Nachbearbeitung erforderlich ist.
- Ausgabe mehrerer Dateien — Hier können Sie die Anzahl der Dateien für Ihre Jobausgabe angeben. Gültige Werte sind 2—999. Es werden möglicherweise weniger Dateien als von Ihnen angegeben ausgegeben, wenn die Spaltenpartitionierung verwendet wird oder wenn die Anzahl der Zeilen in der Ausgabe geringer ist als die Anzahl der von Ihnen angegebenen Dateien.

c. (Optional) Wählen Sie die Spaltenpartitionierung für die Ausgabe von Rezept-Jobs.

Die Spaltenpartitionierung bietet eine weitere Möglichkeit, die Ausgabe Ihrer Rezepturaufgabe in mehrere Dateien zu partitionieren. Die Spaltenpartitionierung kann mit einer neuen oder vorhandenen Amazon S3 S3-Ausgabe oder mit der neuen Amazon S3 S3-Ausgabe von Data Catalog verwendet werden. Es kann nicht mit vorhandenen Amazon S3 S3-Tabellen von Data Catalog verwendet werden. Die Ausgabedateien basieren auf den Werten der von Ihnen angegebenen Spaltennamen. Wenn die von Ihnen angegebenen Spaltennamen eindeutig sind, basieren die resultierenden Amazon S3 S3-Ordnerpfade auf der Reihenfolge der Spaltennamen.

Ein Beispiel für die Partitionierung von Spalten finden Sie Beispiel für die Partitionierung von Spalten im Folgenden.

7. (Optional) Wählen Sie Verschlüsselung für Job-Ausgabe aktivieren aus, um die Job-Ausgabe zu verschlüsseln, die an Ihren Ausgabespeicherort DataBrew schreibt, und wählen Sie dann die Verschlüsselungsmethode aus:

   - SSE-S3 Verschlüsselung verwenden — Die Ausgabe wird mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln verschlüsselt.
   - Verwenden AWS Key Management Service(AWS KMS) — Die Ausgabe wird verschlüsselt mit.AWS KMS Um diese Option zu verwenden, wählen Sie den Amazon-Ressourcennamen (ARN) des AWS KMS Schlüssels, den Sie verwenden möchten. Wenn Sie keinen AWS KMS Schlüssel haben, können Sie einen erstellen, indem Sie Create an AWS KMS key wählen.

8. Wählen Sie für Zugriffsberechtigungen eine AWS Identity and Access Management(IAM-) Rolle aus, mit der Sie in Ihren Ausgabespeicherort schreiben können DataBrew . Für einen Standort, der Ihrem AWS Konto gehört, können Sie die vom `AwsGlueDataBrewDataAccessRole` Dienst verwaltete Rolle wählen. Auf diese Weise können Sie DataBrew auf AWS Ressourcen zugreifen, die Ihnen gehören.

9. Im Bereich Erweiterte Jobeinstellungen können Sie weitere Optionen für die Ausführung Ihres Jobs auswählen:

   - Maximale Anzahl von Einheiten — DataBrew verarbeitet Jobs unter Verwendung mehrerer Rechenknoten, die parallel ausgeführt werden. Die Standardanzahl von Knoten ist 5. Die maximale Anzahl von Knoten ist 149.
   - Job-Timeout — Wenn die Ausführung eines Jobs länger als die von Ihnen hier festgelegte Anzahl von Minuten dauert, schlägt er mit einem Timeout-Fehler fehl. Der Standardwert ist 2.880 Minuten oder 48 Stunden.
   - Anzahl der Wiederholungen — Wenn ein Job während der Ausführung fehlschlägt, DataBrew kann versucht werden, ihn erneut auszuführen. Standardmäßig wird der Job nicht erneut versucht.
   - Amazon CloudWatch Logs für Job aktivieren — Ermöglicht DataBrew die Veröffentlichung von Diagnoseinformationen in CloudWatch Logs. Diese Protokolle können zur Fehlerbehebung oder für weitere Informationen zur Verarbeitung des Jobs nützlich sein.

10. Bei Zeitplan-Jobs können Sie einen DataBrew Job-Zeitplan anwenden, sodass Ihr Job zu einem bestimmten Zeitpunkt oder in regelmäßigen Abständen ausgeführt wird. Weitere Informationen finden Sie unter Automatisieren von Jobläufen mit einem Zeitplan.

11. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Job erstellen. Oder, wenn Sie den Job sofort ausführen möchten, wählen Sie Job erstellen und ausführen.

Sie können den Fortschritt Ihres Jobs überwachen, indem Sie dessen Status überprüfen, während der Job ausgeführt wird. Wenn die Auftragsausführung abgeschlossen ist, ändert sich der Status in Erfolgreich. Die Jobausgabe ist jetzt an dem von Ihnen ausgewählten Ausgabeort verfügbar.

DataBrew speichert Ihre Jobdefinition, sodass Sie denselben Job später ausführen können. Um einen Job erneut auszuführen, wählen Sie Jobs im Navigationsbereich aus. Wählen Sie den Job aus, mit dem Sie arbeiten möchten, und klicken Sie dann auf Job ausführen.

Beispiel für die Partitionierung von Spalten

Gehen Sie als Beispiel für die Spaltenpartitionierung davon aus, dass Sie drei Spalten angeben, von denen jede Zeile einen von zwei möglichen Werten enthält. Die `Dept` Spalte kann den Wert `Admin` oder `Eng` haben. Die `Staff-type` Spalte kann den Wert `Part-time` oder haben`Full-time`. Die `Location` Spalte kann den Wert `Office1` oder haben`Office2`. Die Amazon S3 S3-Buckets für Ihre Jobausgabe sehen ungefähr wie folgt aus.

```
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
```

Automatisieren von Jobläufen mit einem Zeitplan

Sie können DataBrew Jobs jederzeit erneut ausführen und auch DataBrew Jobausführungen mit einem Zeitplan automatisieren.

Um einen Job erneut auszuführen DataBrew

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole unter https://console.aws.amazon.com/databrew/.
2. Wählen Sie im Navigationsbereich Jobs aus. Wählen Sie den Job aus, den Sie ausführen möchten, und klicken Sie dann auf Job ausführen.

Um einen DataBrew Job zu einem bestimmten Zeitpunkt oder in regelmäßigen Abständen auszuführen, erstellen Sie einen DataBrew Job-Zeitplan. Anschließend können Sie Ihren Job so einrichten, dass er gemäß dem Zeitplan ausgeführt wird.

Um einen DataBrew Job-Zeitplan zu erstellen

1. Wählen Sie im Navigationsbereich der DataBrew Konsole Jobs aus. Wählen Sie die Registerkarte Zeitpläne und dann Zeitplan hinzufügen aus.
2. Geben Sie einen Namen für Ihren Zeitplan ein und wählen Sie dann einen Wert für die Ausführungshäufigkeit:

   - Wiederkehrend — Wählen Sie aus, wie oft der Job ausgeführt werden soll (z. B. alle 12 Stunden). Wählen Sie dann aus, an welchem Tag oder an welchen Tagen der Job ausgeführt werden soll. Optional können Sie die Tageszeit eingeben, zu der der Job ausgeführt wird.
   - Zu einer bestimmten Zeit — Geben Sie die Tageszeit ein, zu der der Job ausgeführt werden soll. Wählen Sie dann aus, an welchem Tag oder an welchen Tagen der Job ausgeführt werden soll.
   - CRON eingeben — Definieren Sie den Job-Zeitplan, indem Sie einen gültigen Cron-Ausdruck eingeben. Weitere Informationen finden Sie unter Mit Cron-Ausdrücken für Rezeptjobs arbeiten.
3. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Save (Speichern) aus.

Um einen Job einem Zeitplan zuzuordnen

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie den Job aus, mit dem Sie arbeiten möchten, und wählen Sie dann für Aktionen die Option Bearbeiten aus. .

3. Wählen Sie im Bereich Jobs planen die Option Zeitplan zuordnen aus. Wählen Sie den Namen des Zeitplans aus, den Sie verwenden möchten.
4. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Save (Speichern) aus.

Mit Cron-Ausdrücken für Rezeptjobs arbeiten

Cron-Ausdrücke verfügen über sechs Pflichtfelder, die durch Leerzeichen voneinander getrennt sind. Die Syntax ist wie folgt.

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

In der vorherigen Syntax werden die folgenden Werte und Platzhalter für die angegebenen Felder verwendet.

Felder	Werte	Platzhalter
Minuten	0-59	, - * /
Stunden	0–23	, - * /
Day-of-month	1-31	, - * ? / L W
Monat	1—12 oder JAN-DEC	, - * /
Day-of-week	1—7 oder SUN-SAT	, - * ? / L
Jahr	1970-2199	, - * /

Verwenden Sie diese Platzhalter wie folgt:

- Das Platzhalterzeichen , (Komma) schließt zusätzliche Werte ein. In `Month` diesem Feld `JAN,FEB,MAR` sind Januar, Februar und März enthalten.
- Der Platzhalter - (mit Gedankenstrich) gibt Bereiche an. In dem `Day` Feld umfasst 1—15 die Tage 1 bis 15 des angegebenen Monats.
- Das Platzhalterzeichen * (Sternchen) steht für alle Werte im Feld. In dem `Hours` Feld schließt ein* jede Stunde ein.

- Das Platzhalterzeichen / (Schrägstrich) steht für schrittweise Steigerungen. In das `Minutes` Feld können Sie angeben, **`1/10`** dass ab der ersten Minute der Stunde jede 10. Minute angegeben werden soll (z. B. die 11., 21. und 31. Minute).
- Das Platzhalterzeichen ? (Fragezeichen) steht für einen Wert. Nehmen wir beispielsweise an, dass Sie in das `Day-of-month` Feld 7 eingeben. Wenn es Ihnen egal war, welcher Wochentag der siebte war, können Sie dann eingeben? auf dem `Day-of-week` Feld.
- Der Platzhalter L im `Day-of-week` Feld `Day-of-month` oder gibt den letzten Tag des Monats oder der Woche an.
- Das Platzhalterzeichen W im Feld `Day-of-month` gibt einen Wochentag an. Im Feld `Day-of-month` gibt den `3W` den Tag an, der dem dritten Tag des Monats am nächsten ist.

Für diese Felder und Werte gelten die folgenden Einschränkungen:

- Es ist nicht möglich, die Felder `Day-of-month` und `Day-of-week` im gleichen Cron-Ausdruck anzugeben. Wenn Sie einen Wert in einem der Felder angeben, müssen Sie in dem anderen Feld ein ? (Fragezeichen) eingeben.
- Cron-Ausdrücke, die zu Raten von mehr als 5 Minuten führen, werden nicht unterstützt.

Wenn Sie einen Zeitplan erstellen, können Sie die folgenden Beispiel-Cron-Strings verwenden.

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0	10	*	*	?	*	Führen Sie jeden Tag um 10:00 Uhr (UTC) aus
15	12	*	*	?	*	Ausführung jeden Tag um 12:15 Uhr (UTC)

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0	18	?	*	MON-FRI	*	Ausführung jeden Montag bis Freitag um 18:00 Uhr (UTC)
0	8	1	*	?	*	Läuft jeden ersten Tag des Monats um 8:00 Uhr (UTC)
0/15	*	*	*	?	*	Ausführung alle 15 Minuten
0/10	*	?	*	MON-FRI	*	Ausführung alle 10 Minuten von Montag bis Freitag

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0/5	8-17	?	*	MON-FRI	*	Ausführung alle 5 Minuten von Montag bis Freitag zwischen 08:00 Uhr und 17:55 Uhr (UTC)

Sie können beispielsweise den folgenden Cron-Ausdruck verwenden, um jeden Tag um 12:15 Uhr UTC einen Job auszuführen.

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

Jobs und Jobpläne löschen

Wenn Sie einen Job oder einen Jobplan nicht mehr benötigen, können Sie ihn löschen.

Einen Auftrag löschen

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie den Job aus, den Sie löschen möchten, und wählen Sie dann für Aktionen die Option Löschen aus. .

Um einen Jobplan zu löschen

1. Wählen Sie im Navigationsbereich Jobs und dann die Registerkarte Zeitpläne aus.
2. Wählen Sie den Zeitplan aus, den Sie löschen möchten, und wählen Sie dann für Aktionen die Option Löschen aus. .

Erstellen und Arbeiten mit AWS Glue DataBrew Jobs profilieren

Profiljobs führen eine Reihe von Bewertungen für einen Datensatz durch und geben die Ergebnisse in Amazon S3 aus. Die Informationen, die bei der Datenprofilerstellung gesammelt werden, helfen Ihnen dabei, Ihren Datensatz zu verstehen und zu entscheiden, welche Schritte zur Datenvorbereitung Sie möglicherweise in Ihren Rezepturjobs ausführen möchten.

Die einfachste Methode, einen Profiljob auszuführen, ist die Verwendung der Standardeinstellungen DataBrew . Sie können Ihren Profiljob vor der Ausführung so konfigurieren, dass er nur die gewünschten Informationen zurückgibt.

Gehen Sie wie folgt vor, um einen DataBrew Profiljob zu erstellen.

Um einen Profiljob zu erstellen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die DataBrew Konsole unter https://console.aws.amazon.com/databrew/.
2. Wählen Sie im Navigationsbereich die Option JOBS aus, wählen Sie die Registerkarte Profiljobs und dann Job erstellen aus.
3. Geben Sie einen Namen für Ihren Job ein und wählen Sie dann Einen Profiljob erstellen aus.
4. Geben Sie für die Jobeingabe den Namen des Datensatzes an, für den ein Profil erstellt werden soll.
5. (Optional) Konfigurieren Sie im Bereich Datenprofilkonfigurationen Folgendes:

   - Konfigurationen auf Datensatzebene — Konfigurieren Sie die Details Ihres Profiljobs für alle Spalten in Ihrem Datensatz.

     Optional können Sie die Funktion aktivieren, doppelte Zeilen im Datensatz zu erkennen und zu zählen. Sie können auch Korrelationsmatrix aktivieren und Spalten auswählen, um zu sehen, wie eng die Werte in mehreren Spalten miteinander verknüpft sind. Einzelheiten zu den Statistiken, die Sie auf Datensatzebene konfigurieren können, finden Sie unterKonfigurierbare Statistiken auf Datensatzebene. Sie können Statistiken auf der DataBrew Konsole oder mithilfe der DataBrew API oder der AWS SDKs konfigurieren.
   - Konfigurationen auf Spaltenebene — Mithilfe der Standardeinstellungen für die Profilkonfiguration können Sie die Spalten auswählen, die in Ihren Profiljob aufgenommen werden sollen. Verwenden Sie „Konfigurationsüberschreibung hinzufügen“, um die Spalten auszuwählen, für die die Anzahl der gesammelten Statistiken begrenzt werden soll, oder

um die Standardkonfiguration bestimmter Statistiken zu überschreiben. Einzelheiten zu den Statistiken, die Sie auf Spaltenebene konfigurieren können, finden Sie unterKonfigurierbare Statistiken auf Spaltenebene. Sie können Statistiken auf der DataBrew Konsole oder mithilfe der DataBrew API oder der AWS SDKs konfigurieren.

Stellen Sie sicher, dass alle von Ihnen angegebenen Konfigurationsüberschreibungen für Spalten gelten, die Sie in Ihren Profiljob aufgenommen haben. Wenn es Konflikte zwischen verschiedenen Überschreibungen gibt, die Sie für eine Spalte konfiguriert haben, hat die letzte widersprüchliche Überschreibung Priorität.

6. (Optional) Sie können Datenqualitätsregeln erstellen und zusätzliche Regelsätze für diesen Datensatz anwenden oder bereits angewendete Regeln entfernen. Weitere Informationen zur Überprüfung der Datenqualität finden Sie unter. Validierung der Datenqualität in AWS Glue DataBrew
7. Im Bereich Erweiterte Auftragseinstellungen können Sie weitere Optionen für die Ausführung Ihres Jobs auswählen:

   - Maximale Anzahl von Einheiten — DataBrew verarbeitet Jobs unter Verwendung mehrerer Rechenknoten, die parallel ausgeführt werden. Die Standardanzahl von Knoten ist 5. Die maximale Anzahl von Knoten ist 149.
   - Job-Timeout — Wenn die Ausführung eines Jobs länger als die von Ihnen hier festgelegte Anzahl von Minuten dauert, schlägt er mit einem Timeout-Fehler fehl. Der Standardwert ist 2.880 Minuten oder 48 Stunden.
   - Anzahl der Wiederholungen — Wenn ein Job während der Ausführung fehlschlägt, DataBrew kann versucht werden, ihn erneut auszuführen. Standardmäßig wird der Job nicht erneut versucht.
   - Amazon CloudWatch Logs für Job aktivieren — Ermöglicht DataBrew die Veröffentlichung von Diagnoseinformationen in CloudWatch Logs. Diese Protokolle können zur Fehlerbehebung oder für weitere Informationen zur Verarbeitung des Jobs nützlich sein.

8. Für Associated Schedule können Sie einen DataBrew Job-Zeitplan anwenden, sodass Ihr Job zu einem bestimmten Zeitpunkt oder in regelmäßigen Abständen ausgeführt wird. Weitere Informationen finden Sie unter Automatisieren von Jobläufen mit einem Zeitplan.
9. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Job erstellen. Oder, wenn Sie den Job sofort ausführen möchten, wählen Sie Job erstellen und ausführen.

Programmgesteuertes Erstellen einer Profiljobkonfiguration in AWS Glue DataBrew

In diesem Abschnitt finden Sie Beschreibungen der Schritte und Funktionen von Profiljobs, die Sie programmgesteuert verwenden können. Sie können sie entweder über AWS Command Line Interface(AWS CLI) oder mithilfe eines der AWS SDKs verwenden.

In einem Profiljob können Sie eine Konfiguration anpassen, um zu steuern, wie Ihr DataBrew Datensatz ausgewertet wird. Sie können die Konfiguration auf einen Datensatz oder auf bestimmte Spalten anwenden. Sie können die Konfiguration erstellen, wenn Sie einen Profiljob erstellen, und sie dann jederzeit aktualisieren.

Eine Profilkonfigurationsstruktur besteht aus vier Teilen:

- ProfileColumns Abschnitt
- DatasetStatisticsConfiguration Abschnitt
- ColumnStatisticsConfigurations Abschnitt
- EntityDetectorConfiguration Abschnitt zur Konfiguration von PII

Im Folgenden sehen Sie ein Beispiel.

```
{
    "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 Abschnitt

Legen Sie im `ProfileColumns` Abschnitt Ihrer Struktur die Spalten aus Ihrem Datensatz fest, die Sie in Ihrem Profiljob auswerten möchten. `ProfileColumns`sist eine Liste von Spaltenselektoren (`Selectors`). Sie können entweder einen Spaltennamen oder einen regulären Ausdruck in einer Spaltenauswahl angeben. Ein Beispiel folgt.

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

Wenn diese `ProfileColumns` Option angegeben ist, werden nur Spalten, deren Namen mit einem Namen oder einem regulären Ausdruck in `ProfileColumns` übereinstimmen, in den Profiljob

aufgenommen. Wenn der Profiljob den Datentyp einer ausgewählten Spalte nicht unterstützt, wird die ausgewählte Spalte während der Jobausführung DataBrew übersprungen.

Wenn nicht ProfileColumns definiert, wertet der Profiljob alle unterstützten Spalten aus. Unterstützte Spalten sind Spalten, die Daten eines unterstützten Datentyps enthalten:`ByteType`,`ShortType`,`IntegerType`,`LongType`,`FloatType`, `DoubleTypeString`, oder. `Boolean`

DatasetStatisticsConfiguration Abschnitt

Im `DatasetStatisticsConfiguration` Abschnitt Ihrer Struktur können Sie eine Konfiguration für Bewertungen zwischen den Spalten erstellen. Die Konfiguration umfasst `IncludedStatistics` und`0verrides`. Ein Beispiel folgt.

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

Sie können Bewertungen auswählen, die Sie haben möchten, indem Sie Bewertungsnamen hinzufügen`IncludedStatistics`. Ein Beispiel folgt.

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

Wenn Sie angeben`IncludedStatistics`, dass nur Bewertungen in der Liste in der Profilaufgabe enthalten sind. Wenn `IncludedStatistics` nicht definiert, führt der Profiljob alle unterstützten Evaluierungen mit Standardeinstellungen aus. Sie können alle Bewertungen ausschließen, indem Sie NONE zu `IncludedStatistics` hinzufügen. Ein Beispiel folgt.

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

Konfigurierbare Statistiken auf Datensatzebene

In dem `DatasetStatisticsConfiguration` Abschnitt Ihrer Struktur unterstützt ein Profiljob die in der folgenden Tabelle aufgeführten Bewertungen.

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
DUPLICATE_ROWS_COUNT	Anzahl doppelter Zeilen im Datensatz	all	Aktivieren	duplizieren RowsCoun	Int
KORRELATION	Korrelationskoeffizient nach Pearson zwischen zwei Spalten	number	Aktivieren	Korrelationen (in jeder ausgewählten Spalte)	Objekt

In `IncludedStatistics` können Sie die Standardeinstellungen jeder Auswertung überschreiben, indem Sie eine Überschreibung hinzufügen. Jede Überschreibung umfasst den Namen einer bestimmten Auswertung und eine Parameterzuordnung.

`DatasetStatisticsConfiguration`In unterstützt ein Profiljob die `CORRELATION` Überschreibung. Diese Überschreibung berechnet den Korrelationskoeffizienten nach Pearson zwischen zwei Spalten aus einer Liste ausgewählter Spalten. In der Standardeinstellung werden die ersten 10 numerischen Spalten ausgewählt. Sie können entweder eine Anzahl von Spalten oder eine Liste von Spaltenselektoren angeben, um die Standardeinstellung zu überschreiben.

`CORRELATION`verwendet diese Parameter:

- `columnNumber`— Die Anzahl der numerischen Spalten. Der Profiljob wählt die ersten n Spalten aus dem Datensatz aus. Dieser Wert sollte größer als 1 sein. `"ALL"`Dient zur Auswahl aller numerischen Spalten.

- `columnSelectors:`— Liste der Spaltenselektoren. Jeder Selektor kann entweder einen Spaltennamen oder einen regulären Ausdruck haben.

Ein Beispiel folgt.

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

ColumnStatisticsConfigurations Abschnitt

Im `ColumnStatisticsConfigurations` Abschnitt Ihrer Struktur können Sie Konfigurationen für bestimmte Spalten erstellen. `ColumnStatisticsConfigurations`ist eine Liste von `ColumnStatisticsConfiguration` Einstellungen. Darin `ColumnStatisticsConfiguration` gibt `Selectors` es eine Liste von Spaltenselektoren und `Statistics` für die Konfiguration von Statistiken. Ein Beispiel folgt.

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

`Selectors`ist eine Liste von Spaltenselektoren. Wie bei `ProfileColumns` können Sie in jedem Spaltenselektor entweder einen Spaltennamen oder einen regulären Ausdruck angeben. Wenn Sie angeben`Selectors`, wird die Spaltenkonfiguration auf Spalten angewendet, die einem beliebigen

Spaltenselektor in entsprechen. `Selectors` Andernfalls wird die Konfiguration auf alle unterstützten Spalten angewendet.

`Statistics`In können Sie die Einstellungen ausgewählter Spalten überschreiben. Wie bei`DatasetStatisticsConfiguration`, `Statistics` hat `IncludedStatistics` und`Overrides`.

Um die gewünschten Bewertungen auszuwählen, fügen Sie Bewertungsnamen zu hinzu`IncludedStatistics`.

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

Wenn Sie angeben`IncludedStatistics`, sind nur die Bewertungen in der Liste in der Profilaufgabe enthalten. Andernfalls führt der Profiljob alle unterstützten Auswertungen mit Standardeinstellungen aus.

Sie können alle Bewertungen ausschließen, indem Sie `NONE` zu hinzufügen`IncludedStatistics`.

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

In einigen Fällen gibt es möglicherweise mehrere Konfigurationen mit unterschiedlichen Konfigurationen`IncludedStatistics`, die Sie auf dieselbe Spalte anwenden können. `ColumnStatisticsConfigurations` In diesen Fällen wählt der Profiljob die letzte Konfiguration aus `ColumnStatisticsConfigurations` und wendet `IncludedStatistics` sie auf die ausgewählte Spalte an. Eine neue Konfiguration überschreibt ältere Konfigurationen.

Konfigurierbare Statistiken auf Spaltenebene

`ColumnStatisticsConfigurations`In unterstützt ein Profiljob die in der folgenden Tabelle aufgeführten Bewertungen.

Ein unterstützter Datentyp `number` in dieser Tabelle bedeutet, dass der Datentyp des Attributs einer der folgenden ist: `ByteTypeShortType`,`IntegerType`,`LongType`,`FloatType`, oder`DoubleType`.

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
–	Name der Spalte.	all	–	Name	Zeichenfolge
–	Datentyp der Spalte.	all	–	type	Zeichenfolge
DISTINCT_VALUES_COUNT	Anzahl unterschiedlicher Werte. Ein eindeutiger Wert ist ein Wert, der mindestens einmal vorkommt.	number/boolean/string	Aktiviert	deutlichValuesCount	Int
ENTROPIE	Entropie (Informationstheorie).	number/boolean/Zeichenfolge	Aktiviert	Entropie	Double
INTER_QUARTILSINTER_RANGE	Bereich zwischen dem 25. und dem 75. Prozent der Zahlen.	number	Aktiviert	Interquartilsbereich	Double
KURTOSE	Kurtosis der Kolumne.	number	Aktiviert	Kurtose	Double
MAX	Maximalwert in der Spalte.	number/string Länge	Aktiviert	max	Int/Double
MAXIMALE_WERTE	Liste der Maximalwerte in der Spalte und ihrer Anzahl.	number	Aktiviert	Höchstwerte	Auflisten

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
MEAN	Mittelwert der Werte in der Spalte.	number/string Länge	Aktiviert	mean	Double
MEDIAN	Median der Werte in der Spalte.	number/string Länge	Aktiviert	median	Double
MEDIANE_ABSOLUTE_ABWEICHUNG	Der Median der absoluten Differenzen zwischen den einzelnen Datenpunkten und dem Median einer numerischen Spalte.	number	Aktiviert	MedianAbsoluteDeviation	Double
MIN	Mindestwert in der Spalte.	number/string Länge	Aktiviert	min	Int/Double
MINDESTWERTE	Liste der Mindestwerte in der Spalte und ihrer Anzahl.	number	Aktiviert	Mindestwerte	Auflisten
ANZAHL FEHLENDER WERTE	Anzahl der fehlenden Werte in der Spalte. Null- und Leerzeichenfolgen werden als fehlend betrachtet.	all	Aktiviert	fehltValuesCount	Int

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
MODE	Der am häufigsten vorkommende Wert in der Spalte. Wenn mehrere Werte so häufig vorkommen, ist der Modus einer dieser Werte.	number/string Länge	Aktiviert	mode	Int/Double
MEISTEN_GÄNGIGSTEN_WERTE	Liste der häufigsten Werte in der Spalte.	number/boolean/string	Aktiviert	die meisten CommonValues	Auflisten
ERKENNUNG VON AUSREISSERN	Erkennt Ausreißer in der Spalte mit dem Z_Score-Algorithmus. Zählen Sie die Anzahl der Ausreißer und extrahieren Sie eine Liste mit Stichproben aus den erkannten Ausreißern.	number/string Länge	Aktiviert	zScoreOutliersCount, zScoreOutliersSample	Int/List

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
PERZENTILE	Perzentilwerte der numerischen Spalte (5%, 25%, 75%, 95%).	number	Aktiviert	Perzentil 5, Perzentil 25, Perzentil 75, Perzentil 95	Double
RANGE	Wertebereich in der Spalte.	number	Aktiviert	range	Int/Double
SCHIEFE	Schiefe der Werte in der Spalte.	number	Aktiviert	Schiefe	Double
STANDARDABWEICHUNG	Unvoreingenommene Standardabweichung der Stichprobe der Werte in der Spalte.	number/string Länge	Aktiviert	Standardabweichung	Double
SUM	Summe der Werte in der Spalte.	number	Aktiviert	sum	Int/Double
UNIQUE_VALUES_COUNT	Anzahl der eindeutigen Werte. Ein eindeutiger Wert bedeutet, dass der Wert nur einmal vorkommt.	number/boolean/Zeichenfolge	Aktiviert	einzigartig ValuesCount	Int

Name der Statistik	Beschreibung	Unterstützte Datentypen	Standardstatus	Attribute des Profilergebnisses	Art des Profilergebnisses
WERTVERTEILUNG	Maß für die Verteilung der Werte in der Spalte nach Bereichen.	number/string Länge	Aktiviert	Verteilung der Werte	Auflisten
VARIANCE	Varianz der Werte in der Spalte.	number	Aktiviert	Varianz	Double
Z_SCORE_DISTRIBUTION	Maß für die Verteilung der Z-Score-Werte von Datenpunkten nach Bereich.	number	Aktiviert	zScoreDistribution	Auflisten
ANZAHL NULLEN	Anzahl der Nullen (0s) in der Spalte.	number	Aktiviert	Anzahl Nullen	Int

In `IncludedStatistics` können Sie die Standardparameter jeder Auswertung überschreiben, indem Sie eine Überschreibung hinzufügen. Jede Überschreibung umfasst den Namen einer bestimmten Auswertung und eine Parameterzuordnung.

Parameter für ColumnStatisticsConfigurations Spalten

`ColumnStatisticsConfigurations`In unterstützt ein Profiljob die folgenden Parameter.

In einigen Fällen gibt es möglicherweise mehrere Konfigurationen mit unterschiedlichen Konfigurationen`IncludedStatistics`, die Sie auf dieselbe Spalte anwenden können. `ColumnStatisticsConfigurations` In diesen Fällen wählt der Profiljob die letzte Konfiguration aus `ColumnStatisticsConfigurations` und wendet `IncludedStatistics` sie auf die ausgewählte Spalte an. Eine neue Konfiguration überschreibt ältere Konfigurationen.

MAXIMUM_VALUES

Listet die Maximalwerte in der numerischen Spalte und ihre Anzahl auf. Die Standardlistengröße ist 5. Sie können die Listengröße überschreiben, indem Sie einen Wert für angeben`sampleSize`.

Einstellungen

`sampleSize`— Die Größe der Liste, die die maximale Anzahl und Anzahl von Werten in der numerischen Spalte enthält. Dieser Wert sollte größer als 0 sein. Wird verwendet`"ALL"`, um alle Werte aufzulisten.

Beispiel

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

MINIMALWERTE

Listet die Mindestwerte in der numerischen Spalte und ihre Anzahl auf. Die Standardlistengröße ist 5. Sie können die Listengröße überschreiben, indem Sie einen Wert für angeben`sampleSize`.

Einstellungen

`sampleSize`— Die Größe der Liste, die die maximale Anzahl und Anzahl von Werten in der numerischen Spalte enthält. Dieser Wert sollte größer als 0 sein. Wird verwendet`"ALL"`, um alle Werte aufzulisten.

Beispiel

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

DIE MEISTEN GEBRÄUCHLICHSTEN WERTE

Listet die häufigsten Werte in der Spalte und ihre Anzahl auf. Die Standardlistengröße ist 50. Sie können die Listengröße überschreiben, indem Sie einen Wert für angeben`sampleSize`.

Einstellungen

`sampleSize`— Die Größe der Liste, die die maximale Anzahl und Anzahl von Werten in der numerischen Spalte enthält. Dieser Wert sollte größer als 0 sein. Wird verwendet`"ALL"`, um alle Werte aufzulisten.

Beispiel

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

OUTLIER_DETECTION

Erkennt Ausreißer in der numerischen Spalte oder Zeichenkettenspalte (basierend auf der Zeichenkettenlänge) mit dem Z_Score-Algorithmus.

Ihr Jobprofil zählt die Anzahl der Ausreißer und generiert eine Beispielliste mit Ausreißern und ihren Z-Werten. Die Stichprobenliste ist nach dem absoluten Wert des Z-Werts sortiert. Die Standardlistengröße ist 50.

Der Z_Score-Algorithmus identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht. Der Standardschwellenwert für Ausreißer ist 3.

Sie können einen weiteren Schwellenwert, einen milden Schwellenwert, angeben, um weitere Informationen zu erhalten. Ihr milder Schwellenwert sollte unter Ihrem Schwellenwert liegen. Diese Funktion ist standardmäßig ausgeschaltet. Wenn ein leichter Schwellenwert angegeben ist, gibt Ihr Profiljob eine weitere Zählung zurück`zScoreMildOutliersCount`.

`zScoreOutliersSample`Kann in diesem Fall auch eine Stichprobe von Ausreißern mit leichtem Schwellenwert enthalten.

Einstellungen

- `threshold`— Der Schwellenwert, der bei der Erkennung von Ausreißern verwendet werden soll. Dieser Wert sollte größer oder gleich 0 sein.
- `mildThreshold`— Der milde Schwellenwert, der bei der Erkennung von Ausreißern verwendet werden soll. Dieser Wert sollte größer oder gleich 0 und kleiner als `threshold` sein.
- `sampleSize`— Die Größe der Liste, die Ausreißer in der Spalte enthält. Wird verwendet`"ALL"`, um alle Werte aufzulisten.

Beispiel

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

VALUE_DISTRIBUTION

Misst die Verteilung der Werte in der Spalte anhand der Wertebereiche. Ein Jobprofil gruppiert Werte aus einer numerischen Spalte oder Zeichenkettenspalte (basierend auf der Länge der Zeichenfolge) in Abschnitte nach numerischen Bereichen und generiert eine Liste von Abschnitten. Abschnitte sind aufeinander folgend, und die Obergrenze für einen Bereich ist die Untergrenze für den nächsten Bereich.

Einstellungen

`binNumber`— Anzahl der Fächer. Dieser Wert sollte größer als 0 sein.

Beispiel

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

Z_SCORE_DISTRIBUTION

Misst die Verteilung der Z-Werte von Werten in einer numerischen Spalte. Ein Jobprofil gruppiert Z-Werte von Werten nach numerischen Bereichen in Abschnitte und generiert eine Liste von Abschnitten. Abschnitte sind aufeinander folgend, und die Obergrenze für einen Bereich ist die Untergrenze für den nächsten Bereich.

Einstellungen

`binNumber`— Anzahl der Fächer. Dieser Wert sollte größer als 0 sein.

Beispiel

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

EntityDetectorConfiguration Abschnitt zur Konfiguration von PII

Im `EntityDetectorConfiguration` Abschnitt Ihrer Struktur können Sie die Entitätstypen in Ihrem Datensatz konfigurieren, die Sie als persönlich identifizierbare Informationen (PII) für eine Profilstelle erkennen möchten DataBrew .

EntityTypes

Sie konfigurieren die Entitätstypen DataBrew , die Sie als personenbezogene Daten für Ihre Profilstelle erkennen möchten. Wenn undefiniert `EntityDetectorConfiguration` ist, ist die Entitätserkennung deaktiviert. Die folgenden Entitätstypen können in Ihrem Datensatz erkannt werden:

- USA_SSN
- EMAIL
- USA_ITIN
- USA_PASSPORT_NUMBER
- PHONE_NUMBER
- USA_DRIVING_LICENSE
- BANK_ACCOUNT
- CREDIT_CARD
- IP_ADDRESS
- MAC_ADRESS
- 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

Die Entitätstypgruppe `USA_ALL` wird ebenfalls unterstützt und umfasst alle oben genannten Entitätstypen außer `PERSON_NAME` und`DATE`.

Der Typ von `EntityTypes` ist ein Array von Zeichenketten.

AllowedStatistics

Konfigurieren Sie die Statistiken, die für Spalten ausgeführt werden dürfen, die erkannte Entitäten enthalten. Wenn `AllowedStatistics` nicht definiert, werden keine Statistiken für Spalten berechnet, die erkannte Entitäten enthalten. Eine Liste der gültigen Werte Konfigurierbare Statistiken auf Spaltenebene für den Parameter finden Sie unter. `AllowedStatistics`

Der Typ von `AllowedStatistics` ist ein Array von `AllowedStatistics` Objekten.

Sicherheit bei AWS Glue DataBrew

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud —AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt.AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-partyPrüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS Glue DataBrew, finden Sie unter AWS Dienstleistungen im Bereich nach Compliance-ProgrammenAWS unter .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können AWS Glue DataBrew. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen DataBrew , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer DataBrew Ressourcen unterstützen.

Topics

- Datenschutz in AWS Glue DataBrew
- Identitäts- und Zugriffsmanagement für AWS Glue DataBrew
- Anmeldung und Überwachung DataBrew
- Überprüfung der Einhaltung der Vorschriften für AWS Glue DataBrew
- Resilienz in AWS Glue DataBrew
- Sicherheit der Infrastruktur in AWS Glue DataBrew
- Konfiguration und Schwachstellenanalyse in AWS Glue DataBrew

Datenschutz in AWS Glue DataBrew

DataBrew bietet verschiedene Funktionen, die zum Schutz Ihrer Daten beitragen sollen.

Themen

- Verschlüsselung im Ruhezustand
- Verschlüsselung während der Übertragung
- Schlüsselverwaltung
- Identifizierung und Umgang mit personenbezogenen Daten (PII)
- DataBrew Abhängigkeit von anderen AWS service

Für den Datenschutz in AWS Glue DataBrew gilt das Modell der geteilten Verantwortung das von AWS. Wie in diesem Modell beschrieben,AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz . Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management(IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.

- Wenn Sie für den Zugriff AWSüber eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API DataBrew oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden.AWS CLIAWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

DataBrew unterstützt Datenverschlüsselung im Ruhezustand für DataBrew Projekte und Jobs. Projekte und Jobs können verschlüsselte Daten lesen, und Jobs können verschlüsselte Daten schreiben, indem sie AWS Key Management Service(AWS KMS) aufrufen, um Schlüssel zu generieren und Daten zu entschlüsseln. Sie können KMS-Schlüssel auch verwenden, um die Auftragsprotokolle zu verschlüsseln, die von DataBrew Aufträgen generiert werden. Sie können Verschlüsselungsschlüssel mithilfe der DataBrew Konsole oder der DataBrew API angeben.

> **Important**
>
> AWS Glue DataBrew unterstützt nur symmetrische AWS KMS-Schlüssel. Weitere Informationen finden Sie unter AWS KMS-Schlüssel im AWS Key Management Service Entwicklerhandbuch.

Wenn Sie Jobs DataBrew mit aktivierter Verschlüsselung erstellen, können Sie in der DataBrew Konsole S3-managed serverseitige Verschlüsselungsschlüssel (SSE-S3) oder in () gespeicherte KMS-Schlüssel angeben, um Daten im AWS KMS Ruhezustand zu verschlüsseln. SSE-KMS

> **Important**
>
> Wenn Sie einen Amazon Redshift Redshift-Datensatz verwenden, werden Objekte, die in das bereitgestellte temporäre Verzeichnis entladen wurden, mit verschlüsselt. SSE-S3

Verschlüsselung von Daten, die von Jobs geschrieben wurden DataBrew

DataBrew Jobs können in verschlüsselte Amazon S3 S3-Ziele und verschlüsselte CloudWatch Amazon-Logs schreiben.

Themen

- Einrichtung für DataBrew die Verwendung von Verschlüsselung
- Eine Route erstellen zu AWS KMS für VPC-Jobs
- Verschlüsselung einrichten mit AWS KMS-Schlüssel

Einrichtung für DataBrew die Verwendung von Verschlüsselung

Gehen Sie wie folgt vor, um Ihre DataBrew Umgebung für die Verwendung von Verschlüsselung einzurichten.

So richten Sie Ihre DataBrew Umgebung für die Verwendung von Verschlüsselung ein

1. Erstellen oder aktualisieren Sie Ihre AWS KMS-Schlüssel, um den AWS Identity and Access Management(IAM-) Rollen, die an DataBrew Jobs übergeben werden,AWS KMS Berechtigungen zu erteilen. Diese IAM-Rollen werden verwendet, um CloudWatch Logs und Amazon S3 S3-Ziele zu verschlüsseln. Weitere Informationen finden Sie unter Verschlüsseln von Protokolldaten in CloudWatch Logs Using AWS KMS im Amazon CloudWatch Logs-Benutzerhandbuch.

   Im folgenden Beispiel *"role3"* sind *"role1""role2"*, und IAM-Rollen, die an Jobs übergeben DataBrew werden. Diese Richtlinienerklärung beschreibt eine KMS-Schlüsselrichtlinie, die den aufgelisteten IAM-Rollen die Erlaubnis erteilt, mit diesem KMS-Schlüssel zu verschlüsseln und zu entschlüsseln.

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

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

Die `Service` Anweisung, dargestellt als, ist erforderlich`"Service": "logs.region.amazonaws.com"`, wenn Sie den Schlüssel zum Verschlüsseln von Protokollen verwenden. CloudWatch

2. Stellen Sie sicher, dass der AWS KMS Schlüssel auf eingestellt ist, `ENABLED` bevor er verwendet wird.

Weitere Informationen zum Angeben von Berechtigungen mithilfe von AWS KMS Schlüsselrichtlinien finden Sie unter Verwenden von Schlüsselrichtlinien in AWS KMS.

Eine Route erstellen zu AWS KMS für VPC-Jobs

Sie können sich direkt mit AWS KMSüber einen privaten Endpunkt in Ihrer Virtual Private Cloud (VPC) verbinden, anstatt sich über das Internet zu verbinden. Wenn Sie einen VPC-Endpunkt verwenden,AWS KMS erfolgt die Kommunikation zwischen Ihrer VPC und dem vollständig innerhalb des AWS Netzwerks.

Sie können einen AWS KMS VPC-Endpunkt innerhalb einer VPC erstellen. Ohne diesen Schritt könnten Ihre DataBrew Jobs mit einem fehlschlagen. `kms timeout` Eine ausführliche Anleitung finden Sie unter Herstellen einer Verbindung zu AWS KMS einem VPC-Endpunkt im AWS Key Management Service Entwicklerhandbuch.

Wenn Sie diese Anweisungen auf der VPC-Konsole befolgen, stellen Sie sicher, dass Sie Folgendes tun:

- Wählen Sie „Privaten DNS-Namen aktivieren“.
- Wählen Sie unter Sicherheitsgruppe die Sicherheitsgruppe (einschließlich einer selbstreferenzierenden Regel) aus, die Sie für Ihren DataBrew Job verwenden, der auf Java Database Connectivity (JDBC) zugreift.

Wenn Sie einen DataBrew Job ausführen, der auf JDBC-Datenspeicher zugreift, muss eine Route zum Endpunkt vorhanden sein. DataBrew AWS KMS Sie können die Route mit einem NAT-Gateway (Network Address Translation) oder mit einem AWS KMS VPC-Endpunkt bereitstellen. Informationen zum Erstellen eines NAT-Gateways finden Sie unter NAT-Gateways im Amazon-VPC-Benutzerhandbuch.

Verschlüsselung einrichten mit AWS KMS-Schlüssel

Wenn Sie die Verschlüsselung für einen Job aktivieren, gilt dies sowohl für Amazon S3 als auch CloudWatch. Die übergebene IAM-Rolle muss über die folgenden AWS KMS Berechtigungen verfügen.

Weitere Informationen finden Sie in den folgenden Themen im Benutzerhandbuch zum Amazon Simple Storage Service:

- Weitere Informationen dazu `SSE-S3` finden Sie unter Schützen von Daten mithilfe von Server-Side Verschlüsselung mit S3-Managed Amazon-Verschlüsselungsschlüsseln (SSE-S3).
- Weitere Informationen `SSE-KMS` finden Sie unter Schützen von Daten durch Server-Side Verschlüsselung mit AWS KMS-verwalteten Schlüsseln (). SSE-KMS

Verschlüsselung während der Übertragung

AWS bietet SSL-Verschlüsselung (Secure Sockets Layer) für Daten während der Übertragung.

DataBrew Unterstützung für JDBC-Datenquellen wird bereitgestellt.AWS Glue Wenn Sie eine Verbindung zu JDBC-Datenquellen herstellen, werden die Einstellungen für Ihre AWS Glue Verbindung DataBrew verwendet, einschließlich der Option SSL-Verbindung erforderlich. Weitere Informationen finden Sie unter AWS Glue Verbindungseigenschaften AWS Glue im AWS Glue Entwicklerhandbuch.

AWS KMS bietet sowohl „Bring Your Own Key“ -Verschlüsselung als auch serverseitige Verschlüsselung für die DataBrew Extraktions-, Transformations- und Ladeverarbeitung (ETL) und für die AWS Glue Data Catalog.

Schlüsselverwaltung

Sie können IAM with verwenden, DataBrew um Benutzer,AWS Ressourcen, Gruppen, Rollen und detaillierte Richtlinien für Zugriff, Verweigerung und mehr zu definieren.

Sie können den Zugriff auf die Metadaten je nach den Anforderungen Ihrer Organisation sowohl mit ressourcen- als auch mit identitätsbasierten Richtlinien definieren. Resource-based In den Richtlinien werden die Hauptbenutzer aufgeführt, denen der Zugriff auf Ihre Ressourcen gewährt oder verweigert wird, sodass Sie Richtlinien wie den kontoübergreifenden Zugriff einrichten können. Identitätsrichtlinien sind speziell für Benutzer, Gruppen und Rollen in IAM angefügt.

DataBrew unterstützt die Erstellung Ihrer eigenen AWS KMS key„Bring Your Own Key“ - Verschlüsselung. DataBrew bietet auch serverseitige Verschlüsselung mit KMS-Schlüsseln von AWS KMS for DataBrew jobs.

Identifizierung und Umgang mit personenbezogenen Daten (PII)

Wenn Sie Analysefunktionen oder Modelle für maschinelles Lernen entwickeln, benötigen Sie Sicherheitsvorkehrungen, um die Offenlegung personenbezogener Daten (PII) zu verhindern. PII sind personenbezogene Daten, die zur Identifizierung einer Person verwendet werden können, z. B. eine Adresse, Bankkontonummer oder Telefonnummer. Wenn Datenanalysten und Datenwissenschaftler beispielsweise Datensätze verwenden, um allgemeine demografische Informationen zu ermitteln, sollten sie keinen Zugriff auf die personenbezogenen Daten bestimmter Personen haben.

DataBrew bietet Mechanismen zur Datenmaskierung, um PII-Daten während der Datenaufbereitung zu verschleiern. Je nach den Anforderungen Ihres Unternehmens stehen verschiedene Mechanismen zur Schwärzung personenbezogener Daten zur Verfügung. Sie können die PII-Daten verschleiern, sodass Benutzer sie nicht rückgängig machen können, oder Sie können die Verschleierung rückgängig machen.

Um personenbezogene Daten zu identifizieren und zu maskieren, müssen Sie eine Reihe von Transformationen DataBrew erstellen, mit denen Kunden PII-Daten unkenntlich machen können. Teil dieses Prozesses ist die Bereitstellung von PII-Datenerkennung und Statistiken im Dashboard mit der Übersicht über das Datenprofil auf der Konsole. DataBrew

Sie können die folgenden Techniken zur Datenmaskierung verwenden:

- Substitution — Ersetzen Sie PII-Daten durch andere authentisch aussehende Werte.
- Mischen — Mischen Sie den Wert aus derselben Spalte in verschiedenen Zeilen.

- Deterministische Verschlüsselung — Wenden Sie deterministische Verschlüsselungsalgorithmen auf die Spaltenwerte an. Deterministische Verschlüsselung erzeugt immer denselben Chiffretext für einen Wert.
- Probabilistische Verschlüsselung — Wenden Sie probabilistische Verschlüsselungsalgorithmen auf die Spaltenwerte an. Probabilistische Verschlüsselung erzeugt bei jeder Anwendung einen anderen Chiffretext.
- Entschlüsselung — Entschlüsseln Sie Spalten anhand von Verschlüsselungsschlüsseln.
- Nullstellen oder Löschen — Ersetzen Sie ein bestimmtes Feld durch einen Nullwert oder löschen Sie die Spalte.
- Ausblenden — Verwenden Sie Zeichenverschlüsselung oder maskieren Sie bestimmte Teile in den Spalten.
- Hashing — Wenden Sie Hashfunktionen auf die Spaltenwerte an.

Weitere Informationen zur Verwendung von Transformationen finden Sie unter Rezeptschritte für personenbezogene Daten (PII). Weitere Informationen zur Verwendung von Profiljobs zur Erkennung personenbezogener Daten, einschließlich einer Liste der Entitätstypen, die erkannt werden können, finden Sie im EntityDetectorConfiguration Abschnitt zur Konfiguration personenbezogener Daten unter Programmgesteuertes Erstellen einer Profiljobkonfiguration.

DataBrew Abhängigkeit von anderen AWS service

Um mit der DataBrew Konsole arbeiten zu können, benötigen Sie ein Mindestmaß an Berechtigungen, um mit den DataBrew Ressourcen für Ihr AWS Konto arbeiten zu können. Zusätzlich zu diesen DataBrew Berechtigungen benötigt die Konsole Berechtigungen der folgenden Dienste:

- CloudWatch Protokolliert die Berechtigungen zum Anzeigen von Protokollen.
- IAM-Berechtigungen zum Auflisten und Übergeben von Rollen.
- Amazon EC2 EC2-Berechtigungen zum Auflisten von VPCs, Subnetzen, Sicherheitsgruppen, Instances und anderen Objekten. DataBrew verwendet diese Berechtigungen, um Amazon EC2 EC2-Elemente wie VPCs einzurichten, wenn Jobs ausgeführt werden DataBrew .
- Amazon S3 S3-Berechtigungen zum Auflisten von Buckets und Objekten.
- AWS Glue Berechtigungen zum Lesen von AWS Glue Schemaobjekten wie Datenbanken, Partitionen, Tabellen und Verbindungen.
- AWS Lake Formation Berechtigungen zur Arbeit mit Lake Formation Data Lakes.

Identitäts- und Zugriffsmanagement für AWS Glue DataBrew

AWS Identity and Access Management(IAM) hilft einem Administrator AWS-Service, den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. DataBrew IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- AWS Glue DataBrew und AWS Lake Formation
- Wie AWS Glue DataBrew funktioniert mit IAM
- Identity-based Richtlinienbeispiele für AWS Glue DataBrew
- AWS verwaltete Richtlinien für AWS Glue DataBrew
- Problembehebung bei Identität und Zugriff in AWS Glue DataBrew

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen.Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center(IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung.

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-KontoRoot-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir

raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Benutzer und Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechselnAWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff,AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind.AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identity-based Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Resource-based Richtlinien

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

DataBrew unterstützt keine ressourcenbasierten Richtlinien.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen.AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

DataBrew unterstützt keine ACLs.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations-Benutzerhandbuch.
- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations-Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

AWS Glue DataBrew und AWS Lake Formation

AWS Glue DataBrew unterstützt AWS Lake Formation Berechtigungen für AWS Glue Data Catalog Tabellen. Wenn ein Datensatz eine AWS Glue Data Catalog Tabelle verwendet, die bei Lake Formation registriert ist, muss die für Projekte oder Jobs bereitgestellte IAM-Rolle über die Berechtigungen DESCRIBE und SELECT Lake Formation für die Tabelle verfügen.

AWS Glue DataBrew unterstützt das Schreiben in AWS Glue Data Catalog Tabellen auf AWS Lake Formation der Grundlage von. Wenn ein DataBrew Job einen Datenkatalog verwendet, der bei Lake Formation registriert ist, muss die für die Jobs bereitgestellte IAM-Rolle über INSERT -,

ALTER - und DELETE-Berechtigungen von Lake Formation für die beteiligten Tabellen verfügen. Die IAM-Rolle muss über `glue:UpdateTable` Berechtigungen sowie über Berechtigungen für den Datenspeicherort verfügen, der mit der Datenkatalogtabelle verknüpft ist.

Wie AWS Glue DataBrew funktioniert mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten DataBrew, sollten Sie wissen, mit welchen IAM-Funktionen Sie verwenden können. DataBrew Einen allgemeinen Überblick darüber, wie DataBrew und andere AWS Dienste mit IAM funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Services That Work with IAM.

Themen

- DataBrew identitätsbasierte Richtlinien
- Resource-based Richtlinien in DataBrew
- DataBrew IAM-Rollen

DataBrew identitätsbasierte Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie festlegen, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. DataBrew unterstützt spezifische Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, eine AWS JSON-Richtlinie kann angeben, welcher Principal Aktionen mit welchen Ressourcen und unter welchen Bedingungen ausführen kann.

Das Action-Element einer JSON-Richtlinie beschreibt die Aktionen, für die Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix DataBrew verwendet:`databrew:`. Um einem Benutzer beispielsweise die Berechtigung zum Ausführen einer Amazon-EC2-Instance mit der Amazon-EC2-`RunInstances`-API-Operation zu erteilen, fügen Sie die Aktion `ec2:RunInstances` in seine Richtlinie ein. Richtlinienerklärungen müssen `Action` entweder ein `NotAction` Oder-Element enthalten. DataBrew definiert einen eigenen Satz von Aktionen, die Aufgaben beschreiben, die Sie damit ausführen können.

Um mehrere -Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie folgendermaßen durch Kommas.

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

Sie können auch Platzhalter (*) verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

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

Eine Liste der DataBrew Aktionen finden Sie AWS Glue DataBrew im IAM-Benutzerhandbuch unter Definierte Aktionen von.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Die folgenden DataBrew APIs unterstützen keine Berechtigungen auf Ressourcenebene:

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

Die DataBrew Datensatzressource hat den folgenden Amazon-Ressourcennamen (ARN).

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

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Um beispielsweise die `i-1234567890abcdef0` Instanz in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN.

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

Um alle Instances anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*).

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

Sie können einige DataBrew Aktionen, z. B. zum Erstellen von Ressourcen, nicht für eine bestimmte Ressource ausführen. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

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

Eine Liste der DataBrew Ressourcentypen und ihrer ARNs finden Sie AWS Glue DataBrew im IAM-Benutzerhandbuch unter Defined by (Ressourcen definiert von). Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Glue DataBrew definierte Aktionen.

Bedingungsschlüssel

DataBrew stellt keine dienstspezifischen Bedingungsschlüssel bereit, unterstützt aber die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Beispiele

Beispiele für DataBrew identitätsbasierte Richtlinien finden Sie unter. Identity-based Richtlinienbeispiele für AWS Glue DataBrew

Resource-based Richtlinien in DataBrew

DataBrew unterstützt keine ressourcenbasierten Richtlinien.

DataBrew IAM-Rollen

Eine IAM-Rolle ist eine Entität in Ihrem AWS Konto, die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit DataBrew

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie AssumeRoleoder aufrufen GetFederationToken.

DataBrew unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-linked Rollen

Service-linked Rollen ermöglichen es AWS Diensten, auf Ressourcen in anderen Diensten zuzugreifen, um eine Aktion in Ihrem Namen auszuführen. Service-linked Rollen werden in Ihrem IAM-Konto angezeigt und gehören dem Dienst. Ein -Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Wählen Sie eine IAM-Rolle in DataBrew

Wenn Sie eine Datensatzressource in erstellen DataBrew, wählen Sie eine IAM-Rolle aus, um den DataBrew Zugriff in Ihrem Namen zu ermöglichen. Wenn Sie zuvor eine Servicerolle oder eine mit

einem Dienst verknüpfte Rolle erstellt haben, wird Ihnen DataBrew eine Liste mit Rollen angezeigt, aus denen Sie wählen können. Stellen Sie sicher, dass Sie je nach Bedarf eine Rolle auswählen, die Lesezugriff auf einen Amazon S3 S3-Bucket oder eine Amazon AWS Glue Data Catalog S3-Ressource ermöglicht.

Identity-based Richtlinienbeispiele für AWS Glue DataBrew

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, DataBrew -Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS APIs AWS-ManagementkonsoleAWS CLI, oder ausführen. Ein Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den -Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Best Practices für Richtlinien
- Verwenden der Konsole DataBrew
- Benutzern die Berechtigung zur Anzeige eigener Berechtigungen erteilen
- Verwaltung von DataBrew Ressourcen auf der Grundlage von Tags

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand DataBrew Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar.AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen

finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z.CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Konsole DataBrew

Um auf die AWS Glue DataBrew Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Informationen zu

den DataBrew Ressourcen in Ihrem AWS Konto aufzulisten und einzusehen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver ist als die erforderlichen Mindestberechtigungen, funktioniert die Konsole für Benutzer oder Rollen mit dieser Richtlinie nicht wie vorgesehen.

Um sicherzustellen, dass Benutzer und Rollen die DataBrew Konsole verwenden können, fügen Sie den Entitäten außerdem die folgende AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

```
AWSDataBrewConsoleAccess
```

Sie müssen Benutzern, die nur die API AWS CLI oder die DataBrew API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die Sie ausführen möchten.

Benutzern die Berechtigung zur Anzeige eigener Berechtigungen erteilen

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder 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": "*"
        }
    ]
}
```

Verwaltung von DataBrew Ressourcen auf der Grundlage von Tags

Sie können Bedingungen in Ihrer identitätsbasierten Richtlinie verwenden, um DataBrew Ressourcen auf der Grundlage von Tags zu verwalten, z. B. um die Ressourcen zu löschen, zu aktualisieren oder zu beschreiben. Das folgende Beispiel zeigt eine Richtlinie, die das Löschen eines Projekts verweigert. Das Löschen wird jedoch nur verweigert, wenn das Projekt-Tag Owner den Wert admin hat. Diese Richtlinie gewährt auch die erforderlichen Berechtigungen, um diese Aktion auf der Konsole zu verweigern.

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"}
         }
      }
   ]
```

```
}
```

Sie können diese Richtlinie den -Benutzern in Ihrem Konto zuweisen. Wenn ein Benutzer namens richard-roe versucht, ein DataBrew Projekt zu löschen, darf die Ressource nicht mit Owner=admin oder owner=admin gekennzeichnet werden. Andernfalls wird dem Benutzer die Erlaubnis verweigert, das Projekt zu löschen. Der Bedingungs-Tag-Schlüssel Owner stimmt sowohl mit Besitzer als auch mit Besitzer überein, da bei Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterschieden wird. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

> **Note**
>
> ListDatasets, ListJobs, ListProjects ListRecipes ListRulesets, und unterstützen ListSchedules keine Tag-basierte Zugriffskontrolle.

AWS verwaltete Richtlinien für AWS Glue DataBrew

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher,AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie unter AWS Verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Dienste fügen einer AWS verwalteten Richtlinie gelegentlich zusätzliche Berechtigungen hinzu, um neue Funktionen zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Es ist sehr wahrscheinlich, dass Dienste eine AWS verwaltete Richtlinie aktualisieren, wenn eine neue Funktion eingeführt wird oder wenn neue Operationen verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die ReadOnlyAccessAWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden

nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und eine Beschreibung der Richtlinien für Jobfunktionen finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien für Jobfunktionen.

DataBrew Aktualisierungen für AWS Verwaltete Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien DataBrew seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst. Abonnieren Sie den RSS-Feed auf der Seite DataBrew Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten. Die verwaltete Richtlinie finden Sie in der AWS IAM-Konsole unter AwsGlueDataBrewFullAccessPolicy.

Änderungen	Beschreibung	Date
AWSGlueDataBrewServiceRole— Leseberechtigung für AWS Glue wurde hinzugefügt.	Dieses Update fügt hinzu`glue:GetCustomEntityType` . Diese Berechtigung ist erforderlich, um AWS Glue DataBrew Profiljobs mit PII-identification aktivierter Option auszuführen.	20. März 2024
AWSGlueDataBrewServiceRole- Leseberechtigung für AWS Glue wurde hinzugefügt.	Dieses Update fügt hinzu`glue:BatchGetCustomEntityTypes` . Diese Berechtigung ist erforderlich, um AWS Glue DataBrew Profiljobs mit PII-identification aktivierter Option auszuführen.	9. Mai 2022
AwsGlueDataBrewFullAccessPolicy- Leseberechtigungen für Amazon Redshift-Data DescribeStatements und Amazon S3 GetLifecycleConfiguration wurden hinzugefügt.	Dieses Update unterstützt zusätzlich `redshift-data:DescribeStatement` die Validierung Ihres SQL bei der Erstellung eines Redshift-based Amazon-Datensatzes. Außerdem	4. Februar 2022

Änderungen	Beschreibung	Date
	wird geprüft`s3:GetLifecycleConfiguration` , ob für das Amazon S3 S3-Bucket-Präfix, das Sie als temporäres Verzeichnis angeben, der Lebenszyklus konfiguriert ist. Darüber hinaus ersetzt diese Änderung die „databrew: *“ -Berechtigungen durch eine explizite Liste von Berechtigungen, die alle DataBrew APIs enthält.	

Änderungen	Beschreibung	Date
AwsGlueDataBrewFullAccessPolicy- Read/write Berechtigungen für AWS Secrets Manager wurden hinzugefügt.	Dieses Update fügt `secretsmanager:CreateSecret` `secretsmanager:GetSecretValue` für ein Geheimnis mit dem Namen `databrew!default` ein Standardgeheimnis zur Verwendung mit DataBrew Transformationen hinzu. Darüber hinaus werden Berechtigungen CreateSecret für Geheimnisse mit `AwsGlueDataBrew-` dem Präfix für die Erstellung von Geheimnissen über die DataBrew Konsole hinzugefügt. GenerateRandom, beschrieben in der AWS Key Management Service API-Referenz, wird verwendet, um eine zufällige Bytefolge zu generieren, die kryptografisch sicher ist.	18. November 2021
AWSGlueDataBrewServiceRole- Read/write Berechtigungen für AWS Secrets Manager wurden hinzugefügt.	Dieses Update fügt `secretsmanager:GetSecretValue` für ein Geheimnis mit dem Namen `databrew!default` ein Standardgeheimnis zur Verwendung mit DataBrew Transformationen hinzu.	18. November 2021

Änderungen	Beschreibung	Date
AwsGlueDataBrewFullAccessPolicy- Read/write Berechtigungen für AWS Secrets Manager wurden hinzugefügt.	Dieses Update fügt `secretsmanager:CreateSecret` `secretsmanager:GetSecretValue` für ein Geheimnis mit dem Namen `databrew!default` ein Standardgeheimnis zur Verwendung mit DataBrew Transformationen hinzu. Darüber hinaus werden Berechtigungen CreateSecret für Geheimnisse mit `AwsGlueDataBrew-` dem Präfix für die Erstellung von Geheimnissen über die DataBrew Konsole hinzugefügt. `kms:GenerateRandom` (`https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html`) wird verwendet, um eine zufällige Byte-Zeichenfolge zu generieren, die kryptografisch sicher ist.	18. November 2021
AWSGlueDataBrewServiceRole- Read/write Berechtigungen für AWS Secrets Manager wurden hinzugefügt.	Dieses Update fügt `secretsmanager:GetSecretValue` für ein Geheimnis mit dem Namen `databrew!default` ein Standardgeheimnis zur Verwendung mit DataBrew Transformationen hinzu.	18. November 2021

Änderungen	Beschreibung	Date
AwsGlueDataBrewFullAccessPolicy- Leseberechtigungen für AWS Glue Katalogdatenbanken und Erstellungsberechtigungen für AWS Glue Katalogtabellen wurden hinzugefügt.	Dieses Update fügt Berechtigungen hinzu, um AWS Glue Katalogdatenbanken aufzulisten und neue Katalogtabellen unter einer vorhandenen Datenbank als Teil der Konfiguration der Ausgabe für DataBrew Jobs zu erstellen.	30. Juni 2021
AwsGlueDataBrewFullAccessPolicy- Read/write Berechtigungen für die AppFlow Amazon-Datensatzfunktion wurden hinzugefügt.	Dieses Update fügt Berechtigungen zum Lesen vorhandener AppFlow Amazon-Flows und Flow-Ausführungen sowie zum Erstellen von Flow-Ausführungen hinzu.	28. April 2021
AwsGlueDataBrewFullAccessPolicy- Leseberechtigungen für Datenbank-Datensätze wurden hinzugefügt.	Dieses Update fügt Berechtigungen zum Lesen vorhandener AWS Glue Verbindungen und zum Erstellen neuer AWS Glue Verbindungen für die Verwendung mit DataBrew hinzu. Um die Konsolenerfahrung beim Erstellen neuer Verbindungen zu vereinfachen, ermöglicht es außerdem die Auflistung von Amazon VPC-Ressourcen und Amazon Redshift Redshift-Clustern. Es gibt auch die Erlaubnis, Geheimnisse aufzulisten, aber nicht zu lesen.AWS Secrets Manager	30. März 2021

Änderungen	Beschreibung	Date
DataBrew hat begonnen, Änderungen zu verfolgen	DataBrew hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	30. März 2021

Problembehebung bei Identität und Zugriff in AWS Glue DataBrew

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit DataBrew und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in DataBrew
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb meiner Umgebung zulassen AWS Konto, um auf meine DataBrew Ressourcen zuzugreifen

Ich bin nicht berechtigt, eine Aktion durchzuführen in DataBrew

Wenn Ihnen AWS-Managementkonsole mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, wenden Sie sich an Ihren Administrator, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` -Benutzer versucht, die Konsole zum Anzeigen von Details zu einem Projekt zu verwenden, jedoch nicht über `databrew:DescribeProject`-Berechtigungen verfügt:

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

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-project`* auf die Ressource `databrew:`*`GetProject`* zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an DataBrew übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in DataBrew auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meiner Umgebung zulassen AWS Konto, um auf meine DataBrew Ressourcen zuzugreifen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen DataBrew unterstützt werden, finden Sie unterWie AWS Glue DataBrew funktioniert mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können,AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen.

- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Anmeldung und Überwachung DataBrew

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit DataBrew und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Punkten leichter debuggen können.AWS bietet verschiedene Tools zur Überwachung Ihrer DataBrew Ressourcen und zur Reaktion auf potenzielle Vorfälle:

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Richtlinie gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden DataBrew. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde DataBrew, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen ermitteln.

Überprüfung der Einhaltung der Vorschriften für AWS Glue DataBrew

Third-party Prüfer bewerten die Sicherheit und Einhaltung von Vorschriften im AWS Glue DataBrew Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Informationen darüber, ob AWS-Service ein Programm AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der AWS Sicherheitsdokumentation.

Resilienz in AWS Glue DataBrew

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones.AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Denn wir empfehlen Ihnen AWS Glue DataBrew, Ihre Jobs so zu konfigurieren, dass sie eine oder mehrere Wiederholungen verwenden. Die Anzahl der Wiederholungen für einen Job wird in der DataBrew Konsole unter Erweiterte Auftragseinstellungen konfiguriert.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Sicherheit der Infrastruktur in AWS Glue DataBrew

Als Teil eines verwalteten Services AWS Glue DataBrew wird es durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Für den Zugriff DataBrew über das Netzwerk verwenden Sie AWS veröffentlichte API-Aufrufe. Kunden müssen Transport Layer Security (TLS) 1.0 oder neuer unterstützen. Wir empfehlen TLS 1.2 oder neuer. Clients müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral (DHE) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Diffie-Hellman Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS -Security-Token-Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Themen

- Verwenden AWS Glue DataBrew mit deiner VPC
- Verwenden AWS Glue DataBrew mit VPC-Endpunkten

Verwenden AWS Glue DataBrew mit deiner VPC

Wenn Sie Amazon VPC zum Hosten Ihrer AWS Ressourcen verwenden, können Sie konfigurieren, dass der Datenverkehr AWS Glue DataBrew auf Basis des Amazon VPC-Service über Ihre Virtual Private Cloud (VPC) weitergeleitet wird. DataBrew stellt dazu zunächst eine elastic network interface in dem von Ihnen angegebenen Subnetz bereit. DataBrew hängt dann die von Ihnen angegebene Sicherheitsgruppe an diese Netzwerkschnittstelle an, um den Zugriff zu kontrollieren. Die angegebene Sicherheitsgruppe muss über selbstreferenzierende Regeln für eingehenden und ausgehenden Datenverkehr verfügen. Außerdem müssen für Ihre VPC DNS-Hostnamen und -Auflösung aktiviert sein. Weitere Informationen finden Sie unter Einrichten einer VPC für die Connect zu JDBC-Datenspeichern im AWS Glue Entwicklerhandbuch.

Für AWS Glue Data Catalog Datasets werden VPC-Informationen konfiguriert, wenn Sie eine AWS Glue Verbindung im Datenkatalog erstellen. Um Datenkatalogtabellen für diese Verbindung zu erstellen, führen Sie einen Crawler von der Konsole aus.AWS Glue Weitere Informationen finden Sie unter Auffüllen des AWS Glue Data Catalog im AWS Glue Entwicklerhandbuch.

Geben Sie für Datenbank-Datasets Ihre VPC-Informationen an, wenn Sie die Verbindung von der DataBrew Konsole aus herstellen.

Für die Verwendung AWS Glue DataBrew mit einem VPC-Subnetz ohne NAT benötigen Sie einen Gateway-VPC-Endpunkt zu Amazon S3 und einen VPC-Endpunkt für die Schnittstelle.AWS Glue Weitere Informationen finden Sie unter Erstellen eines Gateway-Endpunkts und Interface VPC-Endpoints (AWS PrivateLink) in der Amazon VPC-Dokumentation. Die von bereitgestellte elastische Schnittstelle DataBrew hat keine öffentliche IPv4-Adresse und unterstützt daher nicht die Verwendung eines VPC-Internet-Gateways.

Endpunkte der Amazon S3 S3-Schnittstelle werden derzeit nicht unterstützt. Wenn Sie Ihr Geheimnis speichern AWS Secrets Manager möchten, benötigen Sie eine Route zum Secrets Manager. Wenn Sie Verschlüsselung verwenden, benötigen Sie eine Route zu AWS Key Management Service(AWS KMS).

Verwenden AWS Glue DataBrew mit VPC-Endpunkten

Wenn Sie Amazon VPC zum Hosten Ihrer AWS Ressourcen verwenden, können Sie eine private Verbindung zwischen Ihrer VPC herstellen und einen DataBrew VPC-Endpunkt bereitstellen. Mit diesem VPC-Endpunkt können Sie DataBrew API-Aufrufe tätigen.

Für die Verwendung DataBrew mit Ihrer DataBrew VPC ist kein VPC-Endpunkt erforderlich. Weitere Informationen finden Sie unter Verwenden AWS Glue DataBrew mit deiner VPC.

Sie können es AWS Glue mit VPC-Endpunkten in allen AWS Regionen verwenden, die beide unterstützen,AWS Glue und VPC-Endpoints.

Weitere Informationen finden Sie unter diesen Themen im Amazon VPC Benutzerhandbuch:

- Was ist Amazon VPC?
- Erstellen eines Schnittstellenendpunkts

Konfiguration und Schwachstellenanalyse in AWS Glue DataBrew

Konfiguration und IT-Kontrollen liegen in der gemeinsamen Verantwortung AWS von Ihnen, unserem Kunden. Weitere Informationen finden Sie im Modell der AWS gemeinsamen Verantwortung.

Überwachen AWS Glue DataBrew

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS Glue DataBrew anderen AWS Lösungen.AWS bietet die folgenden Überwachungstools, mit denen Sie beobachten DataBrew, melden können, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen,AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer Amazon EC2 EC2-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Mit Amazon CloudWatch Events können Sie automatische Benachrichtigungen für bestimmte Ereignisse in einrichten DataBrew. Ereignisse von DataBrew werden nahezu in Echtzeit an CloudWatch Ereignisse übermittelt. Sie können CloudWatch Ereignisse so konfigurieren, dass Ereignisse überwacht und Ziele als Reaktion auf Ereignisse aufgerufen werden, die auf Änderungen Ihrer Ressourcenfreigaben hinweisen. Änderungen an einer Ressourcenfreigabe lösen Ereignisse sowohl für den Eigentümer der Ressourcenfreigabe als auch für die Prinzipale aus, denen Zugriff auf die Ressourcenfreigabe gewährt wurde. Weitere Informationen finden Sie im Amazon CloudWatch Events-Benutzerhandbuch.
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von Amazon EC2 EC2-Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrailerfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden. Der Service gibt die Protokolldateien in einen Amazon S3-Bucket aus, den Sie zuvor angegeben haben. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie im AWS CloudTrail-Benutzerhandbuch.

Themen

- Überwachung DataBrew mit Amazon CloudWatch
- Automatisieren DataBrew mit Ereignissen CloudWatch
- Überwachung DataBrew mit CloudWatch Protokollen
- DataBrew API-Aufrufe protokollieren mit AWS CloudTrail
- Verwenden AWS Benutzerbenachrichtigungen mit AWS Glue Databrew

Überwachung DataBrew mit Amazon CloudWatch

Sie können die DataBrew Nutzung überwachen CloudWatch, wobei Rohdaten gesammelt und zu lesbaren Kennzahlen verarbeitet werden, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

AWS Glue DataBrew meldet die folgenden Metriken im `AWS/DataBrew` Namespace.

Metrik	Description
`SessionCount`	Die Gesamtzahl der DataBrew Sitzungen im Konto des Kunden Gültige Abmessungen: LogGroupName Gültige Statistiken: Summe Einheiten: Anzahl

Automatisieren DataBrew mit Ereignissen CloudWatch

Mit Amazon CloudWatch Events können Sie Ihre AWS Services automatisieren und automatisch auf Systemereignisse wie Probleme mit der Anwendungsverfügbarkeit oder Ressourcenänderungen reagieren. Ereignisse aus AWS Services werden nahezu in Echtzeit an CloudWatch Events übermittelt. Sie können einfache Regeln schreiben, um anzugeben, welche Ereignisse für Sie interessant sind und welche automatisierten Aktionen durchgeführt werden sollen, wenn sich für ein

Ereignis eine Übereinstimmung mit einer Regel ergibt. Die folgenden Aktionen können beispielsweise automatisch ausgelöst werden:

- Aufrufen des Amazon EC2 EC2-Run-Befehls
- Weiterleiten des Ereignisses an Amazon Kinesis Data Streams
- Aktivierung einer Zustandsmaschine AWS Step Functions
- Benachrichtigen eines Amazon SNS-Themas oder einer Amazon SQS-Warteschlange

DataBrew meldet ein Ereignis an CloudWatch Events, wenn sich der Status einer Ressource in Ihrem AWS Konto ändert. Ereignisse werden auf bestmögliche Weise ausgegeben.

Im Folgenden finden Sie Beispiele für mehrere Ereignisse, die verschiedene Status eines DataBrew Jobs zeigen: `SUCCEEDED`,`FAILED`,`TIMEOUT`, und`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"
  }
```

```
}
```

Weitere Informationen finden Sie im Amazon CloudWatch Events-Benutzerhandbuch.

Überwachung DataBrew mit CloudWatch Protokollen

Sie können DataBrew Jobs mithilfe von CloudWatch Logs überwachen. Dabei werden detaillierte Informationen aus dem DataBrew Job-Subsystem gesammelt und zur Überprüfung zur Verfügung gestellt. Diese Protokolle können hilfreich sein, wenn Sie einen Einblick in die Ressourcen erhalten möchten, die Ihr Profil und Ihre Rezepturaufträge verwenden, oder um Probleme zu beheben. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

DataBrew API-Aufrufe protokollieren mit AWS CloudTrail

DataBrew ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden DataBrew. CloudTrail erfasst alle API-Aufrufe DataBrew als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der DataBrew Konsole und Codeaufrufen für die DataBrew API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für DataBrew. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie ermitteln CloudTrail, an welche Anfrage gestellt wurde DataBrew. Sie können auch die IP-Adresse, von der die Anforderung ausging, den Ersteller und den Erstellungszeitpunkt sowie weitere Details bestimmen.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

DataBrew Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet DataBrew, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für DataBrew, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von

Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail-Benutzerhandbuch:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle DataBrew Aktionen werden von der API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von `UpdateRecipe` und `StartJobRun` Aktionen Einträge in den CloudTrail Protokolldateien. `CreateDataset`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Grundlegendes zu DataBrew Einträgen in Protokolldateien

Auch hier handelt es sich bei einem CloudTrail Trail um eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den `CreateProfileJob` Vorgang demonstriert.

```
{
    "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"
}
```

Verwenden AWS Benutzerbenachrichtigungen mit AWS Glue Databrew

Du kannst AWS Benutzerbenachrichtigungen verwenden, um Vertriebskanäle einzurichten, um über AWS Glue Databrew-Ereignisse informiert zu werden. Sie erhalten eine Benachrichtigung, wenn ein Ereignis einer von Ihnen angegebenen Regel entspricht. Sie können Benachrichtigungen für Ereignisse über mehrere Kanäle erhalten, einschließlich E-Mail, Chat-Benachrichtigungen von Amazon Q Developer in Chat-Anwendungen oder AWS Console Mobile Application-Push-Benachrichtigungen. Benachrichtigungen werden auch im Console Notifications Center angezeigt.AWS Benutzerbenachrichtigungen unterstützen die Aggregation, wodurch die Anzahl der Benachrichtigungen, die Sie bei bestimmten Ereignissen erhalten, reduziert werden kann.

Rezeptschritt und Funktionsreferenz

In dieser Referenz finden Sie Beschreibungen der Rezeptschritte und Funktionen, die Sie programmgesteuert verwenden können, entweder über AWS CLI oder mithilfe eines der SDKs.AWS Bei DataBrew einem Rezeptschritt handelt es sich um eine Aktion, mit der Ihre Rohdaten in ein Formular umgewandelt werden, das von Ihrer Datenpipeline verarbeitet werden kann. Eine DataBrew Funktion ist eine spezielle Art von Rezeptschritt, der eine Berechnung auf der Grundlage von Parametern durchführt.

Zu den Kategorien für Transformationen in der Benutzeroberfläche gehören:

- Grundlegende Schritte nach dem Rezept für Spalten
  - Filter
  - Spalte
- Rezeptschritte zur Datenbereinigung
  - Format
  - Bereinigen
  - Extrahieren
- Rezeptschritte zur Datenqualität
  - Fehlen
  - Ungültig
  - Duplikate
  - Ausreißer
- Rezeptschritte für persönlich identifizierbare Informationen (PII)
  - Personenbezogene Daten maskieren
  - Ersetzen Sie persönliche Daten
  - Verschlüsseln Sie persönliche Daten
  - Zeilen mischen
- Rezeptschritte für die Spaltenstruktur
  - Teilen
  - Mischen von
  - Create

- Rezeptschritte zur Spaltenformatierung
  - Dezimale Genauigkeit
  - Tausender-Trennzeichen
  - Zahlen abkürzen
- Rezeptschritte für die Datenstruktur
  - Nest-Unnest
  - Pivot
  - Group (Gruppieren)
  - Join
  - Union
- Rezeptschritte für Datenwissenschaft
  - Text
  - Skalieren
  - Mapping (Zuordnung)
  - Codierung
- Funktionen
  - Mathematische Funktionen
  - Aggregationsfunktionen
  - Textfunktionen
  - Datums- und Zeitfunktionen
  - Fensterfunktionen
  - Web-Funktionen
  - Andere Funktionen

Weitere Hinweise zur Verwendung dieser Rezeptschritte und Funktionen in einem Rezept (einschließlich der Verwendung von Bedingungsausdrücken) finden Sie unterDefinition einer Rezeptstruktur.

In den folgenden Abschnitten werden die Rezeptschritte und Funktionen nach ihrer Funktion geordnet beschrieben.

- Grundlegende Schritte für Spaltenrezepte
- Rezeptschritte für die Datenbereinigung
- Rezeptschritte zur Datenqualität
- Rezeptschritte für persönlich identifizierbare Informationen (PII)
- Rezeptschritte zur Erkennung und Behandlung von Ausreißern
- Rezeptschritte für die Spaltenstruktur
- Rezeptschritte zur Spaltenformatierung
- Rezeptschritte für die Datenstruktur
- Rezeptschritte für Datenwissenschaft
- Mathematische Funktionen
- Aggregationsfunktionen
- Textfunktionen
- Datums- und Zeitfunktionen
- Fensterfunktionen
- Web-Funktionen
- Andere Funktionen

Grundlegende Schritte für Spaltenrezepte

Verwenden Sie diese grundlegenden Spaltenrezeptaktionen, um einfache Transformationen an Ihren Daten durchzuführen.

Themen

- CHANGE_DATA_TYPE
- DELETE
- DUPLIKAT
- JSON_TO_STRUCTS
- MOVE_AFTER
- MOVE_BEFORE
- MOVE_TO_END
- MOVE_TO_INDEX

- MOVE_TO_START
- RENAME
- SORT
- TO_BOOLEAN_COLUMN
- TO_DOUBLE_COLUMN
- TO_NUMBER_COLUMN
- TO_STRING_COLUMN

CHANGE_DATA_TYPE

Ändert den Datentyp einer vorhandenen Spalte.

Wenn ein Spaltenwert nicht in den neuen Typ konvertiert werden kann, wird er durch NULL ersetzt. Dies kann passieren, wenn eine Zeichenkettenspalte in eine Ganzzahlspalte konvertiert wird. Beispielsweise wird die Zeichenfolge „123" zu einer Ganzzahl 123, aber die Zeichenfolge „ABC“ kann nicht zu einer Zahl werden, sodass sie durch einen NULL-Wert ersetzt wird.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Neuer Typ der Spalte. Die folgenden Datentypen werden unterstützt:
  - Byte: 1-Byte-Ganzzahlen mit Vorzeichen. Der Zahlenbereich reicht von -128 bis 127.
  - kurz: 2-Byte-Ganzzahlen mit Vorzeichen. Der Zahlenbereich reicht von -32768 bis 32767.
  - int: 4-Byte-Ganzzahlzahlen mit Vorzeichen. Der Zahlenbereich reicht von -2147483648 bis 2147483647.
  - lang: 8-Byte-Ganzzahlzahlen mit Vorzeichen. Der Zahlenbereich reicht von -9223372036854775808 bis 9223372036854775807.
  - float: 4-Byte-Gleitkommazahlen mit einfacher Genauigkeit.
  - double: 8-Byte-Gleitkommazahlen mit doppelter Genauigkeit.
  - Dezimalzahlen: Vorzeichenbehaftete Dezimalzahlen mit insgesamt bis zu 38 Ziffern und 18 Nachkommastellen.
  - Zeichenfolge: Zeichenkettenwerte.
  - Boolean: Der boolesche Typ hat einen von zwei möglichen Werten: `true` und `false` oder `yes` und `no`.

- timestamp: Werte, die die Felder Jahr, Monat, Tag, Stunde, Minute und Sekunde umfassen.
- Datum: Werte, die die Felder Jahr, Monat und Tag umfassen.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
 }
```

DELETE

Entfernt eine Spalte aus dem Datensatz.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

DUPLIKAT

Erstellt eine neue Spalte mit dem anderen Namen, aber mit denselben Daten. Sowohl die alten als auch die neuen Spalten werden im Datensatz beibehalten.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn`— Ein Name für die doppelte Spalte.

Example Beispiel

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

JSON_TO_STRUCTS

Konvertiert eine JSON-Zeichenfolge in statisch typisierte Strukturen. Während der Konvertierung erkennt es das Schema jedes JSON-Objekts und führt sie zusammen, um das allgemeinste Schema zu erhalten, das die gesamte JSON-Zeichenfolge darstellt. Der Parameter „UnnestLevel“ gibt an, wie viele Ebenen von JSON-Objekten in Strukturen konvertiert werden sollen.

Parameters

- `sourceColumns`— Eine Liste von Quellspalten.
- `regexColumnSelector` –Ein regulärer Ausdruck zur Auswahl der Spalten.
- `removeSourceColumn`— Ein boolescher Wert. Wenn `true` dann, entfernen Sie die Quellspalte; andernfalls behalten Sie sie bei.
- `unnestLevel`— Die Anzahl der Ebenen, deren Verschachtelung aufgehoben werden soll.
- `conditionExpressions`— Bedingungsausdrücke.

Example Beispiel

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

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

MOVE_AFTER

Verschiebt eine Spalte an die Position unmittelbar nach einer anderen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn`— Der Name einer anderen Spalte. Die von angegebene Spalte `sourceColumn` wird unmittelbar nach der von angegebenen Spalte verschoben`targetColumn`.

Example Beispiel

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

MOVE_BEFORE

Verschiebt eine Spalte an die Position unmittelbar vor einer anderen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn`— Der Name einer anderen Spalte. Die von angegebene Spalte `sourceColumn` wird unmittelbar nach der von angegebenen Spalte verschoben`targetColumn`.

Example Beispiel

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

MOVE_TO_END

Verschiebt eine Spalte an die Endposition (letzte Spalte) im Datensatz.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

MOVE_TO_INDEX

Verschiebt eine Spalte an eine durch eine Zahl angegebene Position.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetIndex`— Die neue Position für die Spalte. Positionen beginnen mit 0 — 1 bezieht sich also beispielsweise auf die zweite Spalte, 2 bezieht sich auf die dritte Spalte usw.

Example Beispiel

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

MOVE_TO_START

Verschiebt eine Spalte an die Anfangsposition (erste Spalte) im Datensatz.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

RENAME

Erstellt eine neue Spalte mit dem anderen Namen, aber mit denselben Daten. Die alte Spalte wird dann aus dem Datensatz entfernt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

- `targetColumn`— Ein neuer Name für die Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

Sortiert die Daten in einer oder mehreren Spalten eines Datensatzes in aufsteigender, absteigender oder benutzerdefinierter Reihenfolge.

Parameters

- `expressions`— Eine Zeichenfolge, die eine oder mehrere JSON-encoded Zeichenketten enthält, die Sortierausdrücke darstellen.
  - `sourceColumn`— Eine Zeichenfolge, die den Namen einer vorhandenen Spalte enthält.
  - `ordering`— Die Reihenfolge kann entweder AUFSTEIGEND oder ABSTEIGEND sein.
  - `nullsOrdering`— Die Reihenfolge der Nullen kann entweder NULLS_TOP oder NULLS_BOTTOM lauten, sodass Nullwerte oder fehlende Werte am Anfang oder am Ende der Spalte platziert werden.
  - `customOrder`— Eine Liste von Zeichenketten, die eine benutzerdefinierte Reihenfolge für die Sortierung von Zeichenketten definiert. Standardmäßig sind Zeichenketten alphabetisch sortiert.
  - `isCustomOrderCaseSensitive` – Boolesch. Der Standardwert ist `false`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SORT",
```

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

Example Beispiel für eine benutzerdefinierte Sortierreihenfolge

Im folgenden Beispiel hat die CustomOrder-Ausdruckszeichenfolge das Format einer Objektliste. Jedes Objekt beschreibt einen Sortierausdruck für eine Spalte.

```
[
  {
    "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

Ändert den Datentyp einer vorhandenen Spalte in BOOLEAN.

> **Note**
>
> Wir empfehlen, die Rezeptaktion CHANGE_DATA_TYPE anstelle von TO_BOOLEAN_COLUMN zu verwenden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

- `columnDataType`— Ein Wert, `boolean` der sein muss.

Example Beispiel

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

TO_DOUBLE_COLUMN

Ändert den Datentyp einer vorhandenen Spalte in DOUBLE.

> **Note**
>
> Wir empfehlen, die Rezeptaktion CHANGE_DATA_TYPE anstelle von TO_DOUBLE_COLUMN zu verwenden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Ein Wert, `number` der sein muss.

Example Beispiel

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

```
    }
}
```

TO_NUMBER_COLUMN

Ändert den Datentyp einer vorhandenen Spalte in NUMBER.

> **Note**
>
> Wir empfehlen, die Rezeptaktion CHANGE_DATA_TYPE anstelle von TO_NUMBER_COLUMN zu verwenden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Ein Wert, `number` der sein muss.

Example Beispiel

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

TO_STRING_COLUMN

Ändert den Datentyp einer vorhandenen Spalte in STRING.

> **Note**
>
> Wir empfehlen, die Rezeptaktion CHANGE_DATA_TYPE anstelle von TO_STRING_COLUMN zu verwenden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Ein Wert, `string` der sein muss.

Example Beispiel

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

Rezeptschritte für die Datenbereinigung

Verwenden Sie diese Rezeptschritte zur Datenbereinigung, um einfache Transformationen an vorhandenen Daten durchzuführen.

Themen

- CAPITAL_CASE
- FORMAT_DATE
- KLEINGESCHRIEBENES
- GROSSBUCHSTABEN
- SENTENCE_CASE
- ADD_DOUBLE_QUOTES
- ADD_PREFIX
- ADD_SINGLE_QUOTES
- ADD_SUFFIX
- EXTRACT_BETWEEN DELIMITERS
- EXTRACT_BETWEEN POSITIONS
- EXTRACT_PATTERN

- EXTRACT_VALUE
- REMOVE_COMBINED
- ERSETZEN_ZWISCHEN_TRENNZEICHEN
- ERSETZEN_ZWISCHEN_POSITIONEN
- REPLACE_TEXT

CAPITAL_CASE

Ändert jede Zeichenfolge in einer Spalte, sodass jedes Wort groß geschrieben wird. Bei Großbuchstaben wird der erste Buchstabe jedes Worts groß geschrieben und der Rest des Wortes wird in Kleinbuchstaben umgewandelt. Ein Beispiel ist: Der schnelle braune Fuchs sprang über den Zaun.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

FORMAT_DATE

Gibt eine Spalte zurück, in der eine Datumszeichenfolge in einen formatierten Wert umgewandelt wird.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetDateFormat`— Eines der folgenden Datumsformate:

- mm/dd/yyyy
- mm-dd-yyyy
- dd month yyyy
- month yyyy
- dd month

Example Beispiel

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

KLEINGESCHRIEBENES

Ändert jede Zeichenfolge in einer Spalte in Kleinbuchstaben, zum Beispiel: Der schnelle braune Fuchs ist über den Zaun gesprungen

Parameters

- sourceColumn – Der Name einer vorhandenen Spalte.

Example Beispiel

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

GROSSBUCHSTABEN

Ändert jede Zeichenfolge in einer Spalte in Großbuchstaben, zum Beispiel: DER SCHNELLE BRAUNE FUCHS IST ÜBER DEN ZAUN GESPRUNGEN

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

SENTENCE_CASE

Ändert jede Zeichenfolge in einer Spalte in Groß-/Kleinschreibung. In der Groß-/Kleinschreibung wird der erste Buchstabe jedes Satzes groß geschrieben, und der Rest des Satzes wird in Kleinbuchstaben umgewandelt. Ein Beispiel ist: Der schnelle braune Fuchs. Bin rübergesprungen. Der Zaun

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

```
        }
    }
}
```

ADD_DOUBLE_QUOTES

Schließt die Zeichen in einer Spalte in doppelte Anführungszeichen ein.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

ADD_PREFIX

Fügt ein oder mehrere Zeichen hinzu und verkettet sie als Präfix am Anfang einer Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Das Zeichen oder die Zeichen, die am Anfang der Spaltenwerte stehen sollen.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ADD_PREFIX",
        "Parameters": {
            "pattern": "aaa",
```

```
            "sourceColumn": "info_url"
        }
    }
}
```

ADD_SINGLE_QUOTES

Schließt die Zeichen in einer Spalte in einfache Anführungszeichen ein.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

ADD_SUFFIX

Fügt am Ende einer Spalte ein weiteres Zeichen hinzu und verkettet sie als Suffix.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Das Zeichen oder die Zeichen, die am Ende der Spalte platziert werden sollen.

Example Beispiel

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

```
            "pattern": "bbb",
            "sourceColumn": "info_url"
        }
    }
}
```

EXTRACT_BETWEEN DELIMITERS

Erstellt eine neue Spalte auf der Grundlage von Trennzeichen aus den Werten in einer vorhandenen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `startPattern`— Ein regulärer Ausdruck, der das oder die Zeichen angibt, mit denen die durch Trennzeichen getrennten Werte beginnen.
- `endPattern`— Ein regulärer Ausdruck, der das oder die Trennzeichen angibt, mit denen die durch Trennzeichen getrennten Werte enden.

Example Beispiel

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

EXTRACT_BETWEEN POSITIONS

Erstellt eine neue Spalte auf der Grundlage der Zeichenpositionen aus den Werten in einer vorhandenen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `startPosition`— Die Zeichenposition, an der die Extraktion ausgeführt werden soll.
- `endPosition`— Die Zeichenposition, an der die Extraktion beendet werden soll.

Example Beispiel

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

EXTRACT_PATTERN

Erstellt eine neue Spalte auf der Grundlage eines regulären Ausdrucks aus den Werten in einer vorhandenen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `pattern`— Ein regulärer Ausdruck, der angibt, aus welchem oder welchen Zeichen die neue Spalte extrahiert und erstellt werden soll.

Example Beispiel

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

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

EXTRACT_VALUE

Erstellt eine neue Spalte mit einem extrahierten Wert aus einem benutzerdefinierten Pfad. Wenn die Quellspalte vom Typ Map, Array oder Struct ist, sollte jedes Feld im Pfad mit Backticks maskiert werden (z. B. `name`).

Parameters

- `targetColumn`— Der Name der Zielspalte.
- `sourceColumn`— Name der Quellspalte, aus der der Wert extrahiert werden soll.
- `path`— Der Pfad zu dem spezifischen Schlüssel, den der Benutzer extrahieren möchte. Wenn die Quellspalte vom Typ Map, Array oder Struct ist, sollte jedes Feld im Pfad mit Backticks maskiert werden (zum Beispiel `name`).

  Betrachten Sie das folgende Beispiel für Benutzerinformationen:

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

  Im Folgenden finden Sie Beispiele für Pfade, die Sie je nach Typ der Quellspalte angeben würden:

  - Handelt es sich bei der Quellspalte um den Typ Karte, lautet der Pfad zum Extrahieren der privaten Telefonnummer wie folgt:

```
`user`.`phoneNumber`.`home`
```

- Wenn die Quellspalte vom Typ Array ist, lautet der Pfad zum Extrahieren des zweiten Werts für „Staatsbürgerschaft“ wie folgt:

```
`user`.`citizenship`[1]
```

- Wenn die Quellspalte vom Typ struct ist, lautet der Pfad zum Extrahieren der Postleitzahl:

```
`user`.`address`.`zipcode`
```

Example Beispiel

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

REMOVE_COMBINED

Entfernt ein oder mehrere Zeichen aus einer Spalte, je nachdem, was ein Benutzer angibt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `collapseConsecutiveWhitespace`— Wenn`true`, ersetzt zwei oder mehr Leerzeichen durch genau ein Leerzeichen.
- `removeAllPunctuation`— Wenn`true`, entfernt alle der folgenden Zeichen: `. ! , ?`
- `removeAllQuotes`— Wenn`true`, entfernt alle einfachen Anführungszeichen und doppelten Anführungszeichen.
- `removeAllWhitespace`— Wenn`true`, entfernt alle Leerzeichen.

- `customCharacters`— Ein oder mehrere Zeichen, auf die reagiert werden kann.
- `customValue`— Ein Wert, auf den reagiert werden kann.
- `removeCustomCharacters`— Wenn`true`, entfernt alle durch den `customCharacters` Parameter angegebenen Zeichen.
- `removeCustomValue`— Wenn`true`, entfernt alle durch den `customValue` Parameter angegebenen Zeichen.
- `punctuationally`— Wenn`true`, entfernt die folgenden Zeichen, wenn sie am Anfang oder Ende des Werts vorkommen: `. ! , ?`
- `antidisestablishmentarianism`— Wenn`true`, entfernt einfache Anführungszeichen und doppelte Anführungszeichen am Anfang und Ende des Werts.
- `removeLeadingAndTrailingWhitespace`— Wenn`true`, entfernt alle Leerzeichen am Anfang und Ende des Werts.
- `removeLetters`— Wenn`true`, werden alle Groß- und Kleinbuchstaben (`A`durch`Z`; `a` durch) entfernt. `z`
- `removeNumbers`— Wenn`true`, entfernt alle numerischen Zeichen (`0`durch). 9
- `removeSpecialCharacters`— Wenn`true`, entfernt alle der folgenden Zeichen: `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`

Example Beispiele

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

ERSETZEN_ZWISCHEN_TRENNZEICHEN

Ersetzt die Zeichen zwischen zwei Trennzeichen durch benutzerdefinierten Text.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `startPattern`— Zeichen oder ein regulärer Ausdruck, der angibt, wo die Ersetzung beginnen soll.
- `endPattern`— Zeichen oder ein regulärer Ausdruck, der angibt, wo die Ersetzung enden soll.
- `value`— Das oder die Ersatzzeichen, die ersetzt werden sollen.

Example Beispiel

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

ERSETZEN_ZWISCHEN_POSITIONEN

Ersetzt die Zeichen zwischen zwei Positionen durch benutzerdefinierten Text.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

- `startPosition`— Eine Zahl, die angibt, an welcher Zeichenposition in der Zeichenfolge die Ersetzung beginnen soll.
- `endPosition`— Eine Zahl, die angibt, an welcher Zeichenposition in der Zeichenfolge die Ersetzung enden soll.
- `value`— Das oder die Ersatzzeichen, die ersetzt werden sollen.

Example Beispiel

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

REPLACE_TEXT

Ersetzt eine angegebene Zeichenfolge durch eine andere.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Zeichen oder ein regulärer Ausdruck, der angibt, welche Zeichen in der Quellspalte ersetzt werden sollen.
- `value`— Das oder die Ersatzzeichen, die ersetzt werden sollen.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "REPLACE_TEXT",
```

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

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

Rezeptschritte zur Datenqualität

Gehen Sie wie folgt vor, um fehlende Werte aufzufüllen, ungültige Daten zu entfernen oder Duplikate zu entfernen.

Themen

- ADVANCED_DATATYPE_FILTER
- ADVANCED_DATATYPE_FLAG
- DELETE_DUPLICATE_ROWS
- EXTRACT_ADVANCED_DATATYPE_DETAILS
- FILL_WITH_AVERAGE
- FILL_WITH_CUSTOM
- FILL_WITH_EMPTY
- FILL_WITH_LAST_VALID
- FILL_WITH_MEDIAN
- FILL_WITH_MODE
- FILL_WITH_MOST_FREQUENT

- FILL_WITH_NULL
- FILL_WITH_SUM
- FLAG_DUPLICATE_ROWS
- FLAG_DUPLICATES_IN_COLUMN
- GET_ADVANCED_DATATYPE
- REMOVE_DUPLICATES
- REMOVE_INVALID
- REMOVE_MISSING
- REPLACE_WITH_AVERAGE
- REPLACE_WITH_CUSTOM
- ERSETZEN_DURCH_LEER
- ERSETZEN_DURCH_LETZTE_VALIDE
- ERSETZE DURCH_MEDIAN
- ERSETZEN_MIT_MODUS
- ERSETZEN_DURCH_MEISTES_HÄUFIGES_
- ERSETZEN_MIT_NULL
- ERSETZE DURCH_ROLLING_AVERAGE
- REPLACE_WITH_ROLLING_SUM
- REPLACE_WITH_SUM

ADVANCED_DATATYPE_FILTER

Filtert die aktuelle Quellspalte auf der Grundlage der erweiterten Datentyperkennung. Beispiel: Bei einer Spalte, bei der festgestellt DataBrew wurde, dass sie Postleitzahlen enthält, kann diese Transformation die Spalte anhand der Zeitzone filtern. Die Details, die Sie extrahieren können, hängen vom erkannten Muster ab, wie im Folgenden in den Hinweisen beschrieben.

Parameters

- `sourceColumn`— Der Name einer String-Quellspalte.
- `pattern`— Das zu extrahierende Muster.

- `advancedDataType`— Dabei kann es sich um Telefon, Postleitzahl, Datum, Uhrzeit, Bundesland, Kreditkarte, URL, E-Mail, SSN oder Geschlecht handeln.
- `filter values`— Liste der Zeichenkettenwerte, anhand derer der Benutzer die Spalte filtern möchte.
- `strategy`— KEEP_ROWS oder DISCARD_ROWS oder CLEAR_FILTERS oder CLEAR_OTHERS.
- `clearWithEmpty`— `true` Boolescher Wert oder, um Zeilen mit statt mit zu löschen. `false` `empty` `null`

Hinweise

- Wenn Advanced auf Phone DataType gesetzt ist, kann das Muster AREA_CODE, TIME_ZONE oder COUNTRY_CODE lauten.
- Wenn Advanced auf PLZ DataType gesetzt ist, kann das Muster TIME_ZONE, COUNTRY, STATE, CITY, TYPE oder REGION lauten.
- Wenn Advanced auf Datum und Uhrzeit gesetzt DataType ist, kann das Muster DAY, MONTH_NAME, WEEK, QUARTER oder YEAR lauten.
- Wenn Advanced den DataType Wert State hat, kann das Muster TIME_ZONE lauten.
- Wenn Advanced auf Kreditkarte gesetzt DataType ist, kann das Muster LENGTH oder NETWORK lauten.
- Wenn Advanced URL DataType ist, kann das Muster PROTOCOL, TLD oder DOMAIN lauten.

Example Beispiel

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

```
}
```

ADVANCED_DATATYPE_FLAG

Erstellt eine neue Flag-Spalte auf der Grundlage der Werte für die aktuelle Quellspalte. Wenn beispielsweise eine Quellspalte Postleitzahlen enthält, kann diese Transformation verwendet werden, um Werte als `true` oder `false` basierend auf einer bestimmten Zeitzone zu kennzeichnen. Welche Details Sie extrahieren können, hängt vom erkannten Muster ab, wie in den nachfolgenden Hinweisen beschrieben.

Parameters

- `sourceColumn`— Der Name einer String-Quellspalte.
- `pattern`— Das zu extrahierende Muster.
- `targetColumn`— Der Name der Zielspalte.
- `advancedDataType`— Dabei kann es sich um Telefon, Postleitzahl, Datum, Uhrzeit, Bundesland, Kreditkarte, URL, E-Mail, SSN oder Geschlecht handeln.
- `filter values`— Liste der Zeichenkettenwerte, anhand derer der Benutzer die Spalte filtern möchte.
- `trueString`— Der `true` Wert für die Zielspalte.
- `falseString`— Der `false` Wert für die Zielspalte.

Hinweise

- Wenn Advanced auf Phone DataType gesetzt ist, kann das Muster AREA_CODE, TIME_ZONE oder COUNTRY_CODE lauten.
- Wenn Advanced auf PLZ DataType gesetzt ist, kann das Muster TIME_ZONE, COUNTRY, STATE, CITY, TYPE oder REGION lauten.
- Wenn Advanced auf Datum und Uhrzeit gesetzt DataType ist, kann das Muster DAY, MONTH_NAME, WEEK, QUARTER oder YEAR lauten.
- Wenn Advanced den DataType Wert State hat, kann das Muster TIME_ZONE lauten.
- Wenn Advanced auf Kreditkarte gesetzt DataType ist, kann das Muster LENGTH oder NETWORK lauten.
- Wenn Advanced URL DataType ist, kann das Muster PROTOCOL, TLD oder DOMAIN lauten.

Example Beispiel

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

DELETE_DUPLICATE_ROWS

Löscht jede Zeile, die exakt mit einer früheren Zeile im Datensatz übereinstimmt. Das ursprüngliche Vorkommen wird nicht gelöscht, da es nicht mit einer früheren Zeile übereinstimmt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRACT_ADVANCED_DATATYPE_DETAILS

Extrahiert Details für den erweiterten Datentyp. Welche Details Sie extrahieren können, hängt vom erkannten Muster ab, wie in den nachfolgenden Hinweisen beschrieben.

Parameters

- `sourceColumn`— Der Name einer String-Quellspalte.
- `pattern`— Das zu extrahierende Muster.
- `targetColumn`— Der Name der Zielspalte.

- `advancedDataType`— Dabei kann es sich um Telefon, Postleitzahl, Datum, Uhrzeit, Bundesland, Kreditkarte, URL, E-Mail, SSN oder Geschlecht handeln.

Hinweise

- Wenn „Erweitert“ auf Telefon gesetzt DataType ist, kann das Muster AREA_CODE, TIME_ZONE oder COUNTRY_CODE lauten.
- Wenn Advanced auf PLZ DataType gesetzt ist, kann das Muster TIME_ZONE, COUNTRY, STATE, CITY, TYPE oder REGION lauten.
- Wenn Advanced auf Datum und Uhrzeit gesetzt DataType ist, kann das Muster DAY, MONTH_NAME, WEEK, QUARTER oder YEAR lauten.
- Wenn Advanced den DataType Wert State hat, kann das Muster TIME_ZONE lauten.
- Wenn Advanced auf Kreditkarte gesetzt DataType ist, kann das Muster LENGTH oder NETWORK lauten.
- Wenn Advanced URL DataType ist, kann das Muster PROTOCOL, TLD oder DOMAIN lauten.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

FILL_WITH_AVERAGE

Gibt eine Spalte zurück, in der fehlende Daten durch den Durchschnitt aller Werte ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_CUSTOM

Gibt eine Spalte zurück, in der fehlende Daten durch einen bestimmten Wert ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp für die Spalte. Dieser Typ muss`date`,`number`,`boolean`, `unsupportedstring`, oder sein`timestamp`.
- `value`— Der benutzerdefinierte Wert, der eingegeben werden soll. Der Datentyp muss dem Wert entsprechen, für den Sie sich entschieden haben`columnDataType`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

FILL_WITH_EMPTY

Gibt eine Spalte zurück, in der fehlende Daten durch eine leere Zeichenfolge ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
}
```

FILL_WITH_LAST_VALID

Gibt eine Spalte zurück, in der fehlende Daten durch den neuesten gültigen Wert für diese Spalte ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp für die Spalte. Dieser Typ muss`date`,`number`,`boolean`, `unsupportedstring`, oder sein`timestamp`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

FILL_WITH_MEDIAN

Gibt eine Spalte zurück, in der fehlende Daten durch den Median aller Werte ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MODE

Gibt eine Spalte mit fehlenden Daten zurück, die durch den Modus aller Werte ersetzt wurden.

Sie können auch eine Tie-Breaker-Logik festlegen, wenn einige der Werte identisch sind. Betrachten Sie beispielsweise die folgenden Werte:

`1 2 2 3 3 4`

A `modeType` von `MINIMUM` führt `FILL_WITH_MODE` dazu, dass 2 als Moduswert zurückgegeben wird. Wenn `modeType` ja`MAXIMUM`, ist der Modus 3. Für `AVERAGE` ist der Modus 2,5.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `modeType` – Wie man gleiche Werte in den Daten auflöst. Dieser Wert muss`MINIMUM`, `NONEAVERAGE`, oder sein`MAXIMUM`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MOST_FREQUENT

Gibt eine Spalte zurück, in der fehlende Daten durch den häufigsten Wert ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
    }
}
```

FILL_WITH_NULL

Gibt eine Spalte mit Datenwerten zurück, die durch Null ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
    }
}
```

FILL_WITH_SUM

Gibt eine Spalte zurück, in der fehlende Daten durch die Summe aller Werte ersetzt wurden.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FLAG_DUPLICATE_ROWS

Gibt eine neue Spalte mit einem bestimmten Wert in jeder Zeile zurück, der angibt, ob diese Zeile exakt mit einer früheren Zeile in der Datenmenge übereinstimmt. Wenn Übereinstimmungen gefunden werden, werden diese als Duplikate gekennzeichnet. Das ursprüngliche Vorkommen wird nicht gekennzeichnet, da es nicht mit einer früheren Zeile übereinstimmt.

Parameters

- `trueString` – Wert, der eingefügt werden soll, wenn die Zeile mit einer früheren Zeile übereinstimmt.
- `falseString` – Wert, der eingefügt werden soll, wenn die Zeile eindeutig ist.

- `targetColumn` – Name der neuen Spalte, die in den Datensatz eingefügt wird.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

FLAG_DUPLICATES_IN_COLUMN

Gibt eine neue Spalte mit einem bestimmten Wert in jeder Zeile zurück, der angibt, ob der Wert in der Quellspalte der Zeile mit einem Wert in einer früheren Zeile der Quellspalte übereinstimmt. Wenn Übereinstimmungen gefunden werden, werden diese als Duplikate gekennzeichnet. Das ursprüngliche Vorkommen wird nicht gekennzeichnet, da es nicht mit einer früheren Zeile übereinstimmt.

Parameters

- `sourceColumn` – Name der Quellspalte.
- `targetColumn` – Name der Zielspalte.
- `trueString` – Zeichenfolge, die in die Zielspalte eingefügt werden soll, wenn ein Quellspaltenwert einen früheren Wert in dieser Spalte dupliziert.
- `falseString` – Zeichenfolge, die in die Zielspalte eingefügt werden soll, wenn sich ein Quellspaltenwert von früheren Werten in dieser Spalte unterscheidet.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
```

```
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
}
```

GET_ADVANCED_DATATYPE

Identifiziert bei einer gegebenen Zeichenkettenspalte den erweiterten Datentyp der Spalte, falls vorhanden.

Parameters

- `columnName`— Der Name der Zeichenkettenspalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

REMOVE_DUPLICATES

Löscht eine ganze Zeile, wenn in einer ausgewählten Quellspalte ein doppelter Wert gefunden wird.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

```
{
```

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

REMOVE_INVALID

Löscht eine ganze Zeile, wenn in einer Spalte dieser Zeile ein ungültiger Wert gefunden wird.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
}
```

REMOVE_MISSING

Gibt nur die Zeilen zurück, in denen in einer angegebenen Spalte keine Daten fehlen.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

Example Beispiel

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

REPLACE_WITH_AVERAGE

Ersetzt jeden ungültigen Wert in einer Spalte durch den Durchschnitt aller anderen Werte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "age"
        }
    }
}
```

REPLACE_WITH_CUSTOM

Ersetzt erkannte Entitäten durch einen benutzerdefinierten Wert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

- `sourceColumns`— Eine Liste vorhandener Spaltennamen.
- `columnDataType`— Der Datentyp der Spalte.
- `value`— Der benutzerdefinierte Wert, der verwendet werden soll, um ungültige Werte zu ersetzen.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

> **Note**
>
> Verwenden Sie entweder `sourceColumn` oder`sourceColumns`, aber nicht beide.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

ERSETZEN_DURCH_LEER

Ersetzt jeden ungültigen Wert in einer Spalte durch einen leeren Wert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew

können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

ERSETZEN_DURCH_LETZTE_VALIDE

Ersetzt jeden ungültigen Wert in einer Spalte durch den letzten gültigen Wert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
    }
```

```
}
```

ERSETZE DURCH_MEDIAN

Ersetzt jeden ungültigen Wert in einer Spalte durch den Median aller anderen Werte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

ERSETZEN_MIT_MODUS

Ersetzt jeden ungültigen Wert in einer Spalte durch den Modus aller anderen Werte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.
- `modeType` – Wie man gleiche Werte in den Daten auflöst. Dieser Wert muss`MINIMUM`, `NONEAVERAGE`, oder sein`MAXIMUM`.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
    }
}
```

ERSETZEN_DURCH_MEISTES_HÄUFIGES_

Ersetzt jeden ungültigen Wert in einer Spalte durch den häufigsten Spaltenwert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "wind_direction"
        }
    }
}
```

ERSETZEN_MIT_NULL

Ersetzt jeden ungültigen Wert in einer Spalte durch einen Nullwert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte.
- `advancedDataType`— Spezielle Datentypen, die DataBrew in einer Spalte erkannt werden, die den Datentyp hat`string`. Zu den Typen, die in einer `string` Spalte erkannt werden DataBrew können, gehören SSN, E-Mail, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse DateTime, Währung ZipCode, Land, Region, Bundesland und Stadt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

ERSETZE DURCH_ROLLING_AVERAGE

Ersetzt jeden Wert in einer Spalte durch den gleitenden Durchschnitt aus einem vorherigen „Zeilenfenster“.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.
- `period`- — Die Größe des Fensters. Wenn beispielsweise 10 `period` ist, wird der gleitende Durchschnitt anhand der vorherigen 10 Zeilen berechnet.

Example Beispiel

```
{
```

```
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_ROLLING_SUM

Ersetzt jeden Wert in einer Spalte durch die rollierende Summe aus einem vorherigen „Fenster“ von Zeilen.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.
- `period`- — Die Größe des Fensters. Wenn beispielsweise 10 `period` ist, wird die rollierende Summe anhand der vorherigen 10 Zeilen berechnet.

Example Beispiel

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_SUM

Ersetzt jeden ungültigen Wert in einer Spalte durch die Summe aller anderen Werte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `columnDataType`— Der Datentyp der Spalte. Dieser Typ muss sein`number`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

Rezeptschritte für persönlich identifizierbare Informationen (PII)

Verwenden Sie diese Rezeptschritte, um Transformationen an personenbezogenen Daten (PII) in einem Datensatz durchzuführen.

> **Note**
>
> Zusätzlich zu den Rezeptschritten in diesem Abschnitt gibt es DataBrew Rezeptschritte, die nicht speziell für personenbezogene Daten entwickelt wurden und die Sie für den Umgang mit personenbezogenen Daten verwenden können. Ein Beispiel ist ein einfacher Schritt mit einem SpaltenrezeptDELETE, bei dem eine Spalte gelöscht wird.

Themen

- CRYPTOGRAPHIC_HASH
- ENTSCHLÜSSELN

- DETERMINISTIC_DECRYPT
- DETERMINISTIC_ENCRYPT
- VERSCHLÜSSELN
- MASK_CUSTOM
- MASK_DATE
- MASK_DELIMITER
- MASK_RANGE
- ERSETZEN_MIT_RANDOM_BETWEEN
- ERSETZEN_DURCH_ZUFÄLLIGE_DATE_BETWEEN
- SHUFFLE_ROWS

CRYPTOGRAPHIC_HASH

Wendet einen Algorithmus auf Hashwerte in der Spalte an.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `secretId` – Der ARN des geheimen Secrets-Manager-Schlüssels. Der Schlüssel, der im Präfixalgorithmus für den Hash-basierten Nachrichtenauthentifizierungscode (HMAC) verwendet wird, um die Quellspalten zu hashen, oder `databrew!default` ist die Base64-dekodierte Ausgabe für den Wert des geheimen Schlüssels von Secrets Manager.
- `secretVersion` – Optional. Standardmäßig wird die neueste Geheimnisversion verwendet.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.
- `createSecretIfMissing` – Optionaler boolescher Wert. Falls „true“ wird versucht, das Geheimnis im Namen des Aufrufers zu erstellen.
- `algorithm` – Der Algorithmus, der zum Hashing Ihrer Daten verwendet wird. Gültige Aufzählungswerte: MD5, SHA1, SHA256, SHA512, HMAC_MD5, HMAC_SHA1, HMAC_SHA256, HMAC_SHA512

  Jede Option bezieht sich auf einen anderen Hash-Algorithmus. Diese Optionen mit dem Präfix „HMAC“ beziehen sich auf einen verschlüsselten Hash-Algorithmus und erfordern den Parameter. `secretId` Für Optionen ohne das Präfix „HMAC“ ist der `secretId` Parameter nicht erforderlich.

Wenn Sie keinen Hash-Algorithmus angeben, verwendet der Dienst standardmäßig „HMAC_SHA256".

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "entityTypeFilter": ["USA_ALL"]
}
```

Bei der Arbeit in der interaktiven Oberfläche muss der Konsolenbenutzer zusätzlich zur Rolle des Projekts auch über die entsprechende Zugriffsberechtigung für das angegebene Secrets Manager Manager-Geheimnis verfügen. `secretsmanager:GetSecretValue`

Beispielrichtlinie:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

Sie können sich auch dafür entscheiden, das DataBrew-created Standardgeheimnis zu verwenden, indem Sie `databrew!default` als secretId und den Parameter `createSecretIfMissing` als true übergeben. Dies wird für die Produktion nicht empfohlen. Jeder, der diese AwsGlueDataBrewFullAccessPolicyRolle innehat, kann das Standardgeheimnis verwenden.

ENTSCHLÜSSELN

Sie können die DECRYPT-Transformation verwenden, um das Innere von zu entschlüsseln. DataBrew Ihre Daten können auch außerhalb des DataBrew Encryption SDK entschlüsselt werden.AWS Wenn der angegebene KMS-Schlüssel-ARN nicht mit dem übereinstimmt, der zum Verschlüsseln der Spalte verwendet wurde, scheitert der Entschlüsselungsvorgang. Weitere Informationen zum AWS Encryption SDK finden Sie unter Was ist das AWS Encryption SDK im AWS Encryption SDK Entwicklerhandbuch.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `kmsKeyArn`— Der Schlüssel-ARN des AWS Key Management Service-Schlüssels, der zum Entschlüsseln der Quellspalten verwendet werden soll. Weitere Informationen zum Schlüssel-ARN finden Sie unter Schlüssel-ARN im AWS Key Management Service Entwicklerhandbuch.

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
}
```

Bei der Arbeit in der interaktiven Oberfläche muss der Konsolenbenutzer neben der Rolle des Projekts auch über Berechtigungen für `kms:GenerateDataKey` den bereitgestellten KMS-Schlüssel verfügen. `kms:Decrypt`

Beispielrichtlinie:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": [
```

```
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

DETERMINISTIC_DECRYPT

Entschlüsselt mit DETERMINISTIC_ENCRYPT verschlüsselte Daten.

Diese Transformation ist nicht zulässig, wenn die angegebene geheime ID und Version nicht mit dem übereinstimmt, was zum Verschlüsseln der Spalte verwendet wurde.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `secretId`— Der ARN des geheimen Schlüssels von Secrets Manager, der zum Entschlüsseln der Quellspalten verwendet werden soll.
- `secretVersion` – Optional. Standardmäßig wird die neueste Geheimnisversion verwendet.

Beispiel

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

Bei der Arbeit in der interaktiven Oberfläche muss der Konsolenbenutzer zusätzlich zur Rolle des Projekts über die Berechtigung secretsmanager: GetSecretValue für das angegebene Secrets Manager-Geheimnis verfügen.

Beispiel für eine Richtlinie:

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

DETERMINISTIC_ENCRYPT

Verschlüsselt die Spalte mithilfe eines AES-GCM-SIV 256-Bit-Schlüssels. Mit DETERMINISTIC_ENCRYPT verschlüsselte Daten können nur innerhalb der DETERMINISTIC_DECRYPT-Transformation entschlüsselt werden. DataBrew Diese Transformation verwendet AWS KMS nicht das AWS Encryption SDK, sondern die LC-GitHub-Bibliothek.AWS

Kann bis zu 400 KB pro Zelle verschlüsseln. Behält den Datentyp beim Entschlüsseln nicht bei.

> **Note**
>
> Hinweis: Es wird davon abgeraten, ein Geheimnis länger als ein Jahr zu verwenden.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `secretId`— Der ARN des geheimen Schlüssels von Secrets Manager, der zum Verschlüsseln der Quellspalten oder Databrew verwendet werden soll! Standard.
- `secretVersion` – Optional. Standardmäßig wird die neueste Geheimnisversion verwendet.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.
- `createSecretIfMissing` – Optionaler boolescher Wert. Falls „true“ wird versucht, das Geheimnis im Namen des Aufrufers zu erstellen.

Beispiel

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123",
   "entityTypeFilter": ["USA_ALL"]
}
```

Bei der Arbeit in der interaktiven Oberfläche muss der Konsolenbenutzer zusätzlich zur Rolle des Projekts auch über die entsprechende Zugriffsberechtigung für das angegebene Secrets Manager Manager-Geheimnis verfügen. `secretsmanager:GetSecretValue`

Beispiel für eine Richtlinie

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

VERSCHLÜSSELN

Verschlüsselt Werte in den Quellspalten mit dem AWS Encryption SDK. Die DECRYPT-Transformation kann verwendet werden, um innerhalb von zu entschlüsseln. DataBrew Sie können die Daten auch außerhalb des Encryption DataBrew SDK entschlüsseln.AWS

Die ENCRYPT-Transformation kann bis zu 128 MiB pro Zelle verschlüsseln. Es wird versucht, das Format bei der Entschlüsselung beizubehalten. Um den Datentyp beizubehalten, müssen

die Datentyp-Metadaten auf weniger als 1 KB serialisiert werden. Andernfalls müssen Sie den Parameter `preserveDataType` auf „false“ festlegen. Die Datentyp-Metadaten werden im Verschlüsselungskontext im Klartext gespeichert. Weitere Informationen zum Verschlüsselungskontext finden Sie unter Verschlüsselungskontext im AWS Key Management Service Entwicklerhandbuch.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `kmsKeyArn`— Der Schlüssel-ARN des AWS Key Management Service-Schlüssels, der zum Verschlüsseln der Quellspalten verwendet werden soll. Weitere Informationen zum Schlüssel-ARN finden Sie unter Schlüssel-ARN im AWS Key Management Service Entwicklerhandbuch.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.
- `preserveDataType` – Optionaler boolescher Wert. Standardwert ist „true“. Wenn der Wert „false“ ist, wird der Datentyp nicht gespeichert.

Im folgenden Beispiel `preserveDataType` sind `entityTypeFilter` und optional.

Beispiel

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

Bei der Arbeit in der interaktiven Oberfläche muss der Konsolenbenutzer zusätzlich zur Rolle des Projekts auch `kms:GenerateDataKey` über die entsprechenden Berechtigungen für den angegebenen AWS KMS Schlüssel verfügen.

Beispiel für eine Richtlinie:

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678901:key/kms-key-id"
        ]
    }
  ]
}
```

MASK_CUSTOM

Maskiert Zeichen, die einem angegebenen benutzerdefinierten Wert entsprechen.

Parameters

- `sourceColumns`— Eine Liste vorhandener Spaltennamen.
- `maskSymbol`— Ein Symbol, das verwendet wird, um bestimmte Zeichen zu ersetzen.
- `regex`— Falls der Wert wahr ist, wird dies `customValue` als übereinstimmendes Regex-Muster behandelt.
- `customValue`— Alle Vorkommen (oder Regex-Übereinstimmungen) von `customValue` werden in der Zeichenfolge maskiert.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.

Example Beispiel

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
        "Parameters": {
            "sourceColumns": ["company"],
```

```
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
}
```

MASK_DATE

Maskiert Komponenten eines Datums mit einem benutzerdefinierten Maskensymbol.

Parameters

- `sourceColumns`— Eine Liste vorhandener Spaltennamen.
- `maskSymbol`— Ein Symbol, das verwendet wird, um bestimmte Zeichen zu ersetzen.
- `redact`— Ein Array von Aufzählungen von Datumskomponenten, die maskiert werden sollen. Gültige Aufzählungswerte: JAHR, MONAT, TAG, STUNDE, MINUTE, SEKUNDE, MILLISEKUNDE.
- `locale`— Optionales IETF BCP 47-Sprachkennzeichen. Standardeinstellung: `en`. Das für die Datumsformatierung zu verwendende Gebietsschema.

Example Beispiel

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

MASK_DELIMITER

Maskiert Zeichen zwischen zwei Trennzeichen mit einem benutzerdefinierten Maskierungssymbol.

Parameters

- `sourceColumns`— Eine Liste vorhandener Spaltennamen.

- `maskSymbol`— Ein Symbol, das verwendet wird, um bestimmte Zeichen zu ersetzen.
- `startDelimiter`— Ein Zeichen, das angibt, wo die Maskierung beginnen soll. Wenn Sie diesen Parameter weglassen, wird die Maske ab dem Anfang der Zeichenfolge angewendet.
- `endDelimiter`— Ein Zeichen, das angibt, wo die Maskierung enden soll. Wenn Sie diesen Parameter weglassen, wird die Maskierung vom StartDelimiter bis zum Ende der Zeichenfolge angewendet.
- `preserveDelimiters`— Falls wahr, wird eine Maske auf Trennzeichen angewendet.
- `alphabet`— Eine Reihe von Zeichensätzen, die bei der Maskierung beibehalten werden sollen. Gültige Aufzählungswerte: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.

Example Beispiel

```
// 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

Maskiert Zeichen zwischen zwei Positionen mit einem benutzerdefinierten Maskierungssymbol.

Parameters

- `sourceColumns`— Eine Liste vorhandener Spaltennamen.

- `maskSymbol`— Ein Symbol, das verwendet wird, um bestimmte Zeichen zu ersetzen.
- `start`— Eine Zahl, die angibt, an welcher Zeichenposition die Maskierung beginnen soll (0-indexiert, einschließlich). Negative Indizierung ist zulässig. Wenn Sie diesen Parameter weglassen, wird die Maske vom Anfang der Zeichenfolge bis zu 'stop' angewendet.
- `stop`— Eine Zahl, die angibt, an welcher Zeichenposition die Maskierung enden soll (0-indexiert, exklusiv). Negative Indizierung ist zulässig. Wenn Sie diesen Parameter weglassen, wird die Maske von 'Start' bis zum Ende der Zeichenfolge angewendet.
- `alphabet`— Eine Reihe von Zeichensatzaufzählungen, die bei der Maskierung beibehalten werden sollen. Gültige Aufzählungswerte: SYMBOLS, WHITESPACE.
- `entityTypeFilter`— Optionales Array von Entitätstypen. Kann verwendet werden, um nur erkannte personenbezogene Daten (Personally Identifyable Information, PII) in einer Freitextspalte zu verschlüsseln.

Example Beispiel

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

ERSETZEN_MIT_RANDOM_BETWEEN

Ersetzt Werte durch eine Zufallszahl.

Parameters

- `lowerBound`— Die Untergrenze des Zufallszahlenbereichs.
- `sourceColumns`— Eine Liste vorhandener Spaltennamen.
- `upperBound`— Die Obergrenze des Zufallszahlenbereichs.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
    }
}
```

ERSETZEN_DURCH_ZUFÄLLIGE_DATE_BETWEEN

Ersetzt Werte durch ein zufälliges Datum.

Parameters

- `startDate`— Der Beginn des Datumsbereichs, aus dem ein zufälliges Datum ausgewählt wird.
- `sourceColumns`— Eine Liste vorhandener Spaltennamen.
- `endDate`— Das Ende des Datumsbereichs, aus dem ein zufälliges Datum ausgewählt wird.

Example Beispiel

```
{
    "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

Mischt Werte in einer bestimmten Spalte. Die Mischung kann bei Werten erfolgen, die nach einer sekundären Spalte gruppiert sind.

Parameters

- `sourceColumns` – Ein Array vorhandener Spalten.
- `groupByColumns`— Eine Reihe von Spalten, nach denen die Quellspalten beim Mischen gruppiert werden.

Example Beispiel

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
}
```

Rezeptschritte zur Erkennung und Behandlung von Ausreißern

Verwenden Sie diese Rezeptschritte, um mit Ausreißern in Ihren Daten zu arbeiten und erweiterte Transformationen an ihnen durchzuführen.

Themen

- FLAGGENAUSREISSER
- AUSREISSER ENTFERNEN
- AUSREISSER ERSETZEN
- AUSREISSER MIT Z-SCORE NEU SKALIEREN
- AUSREISSER MIT SCHRÄGLAGE NEU SKALIEREN

FLAGGENAUSREISSER

Gibt eine neue Spalte zurück, die in jeder Zeile einen anpassbaren Wert enthält, der angibt, ob der Wert der Quellspalte ein Ausreißer ist.

Parameters

- `sourceColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `targetColumn`— Gibt den Namen einer neuen Spalte an, in die die Ergebnisse der Strategie zur Bewertung von Ausreißern eingefügt werden sollen.
- `outlierStrategy`— Gibt den Ansatz an, der bei der Erkennung von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - `Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht.
  - `MODIFIED_Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die absolute Abweichung vom Median abweicht.
  - `IQR`— Identifiziert einen Wert als Ausreißer, wenn er über das erste und letzte Quartil der Spaltendaten hinausgeht. Der Interquartilsbereich (IQR) gibt an, wo sich die mittleren 50% der Datenpunkte befinden.
- `threshold`— Gibt den Schwellenwert an, der bei der Erkennung von Ausreißern verwendet werden soll. Der `sourceColumn` Wert wird als Ausreißer identifiziert, wenn der Wert, der mit dem berechnet wird, diese Zahl `outlierStrategy` überschreitet. Die Voreinstellung ist 3.
- `trueString`— Gibt den Zeichenkettenwert an, der verwendet werden soll, wenn ein Ausreißer erkannt wird. Die Standardeinstellung ist „True“.
- `falseString`— Gibt den Zeichenkettenwert an, der verwendet werden soll, wenn kein Ausreißer erkannt wird. Die Standardeinstellung ist „False“.

Die folgenden Beispiele zeigen die Syntax für eine einzelne RecipeActionOperation. Ein Rezept enthält mindestens eine RecipeStepOperation, und ein Rezeptschritt enthält mindestens eine Rezeptaktion. Eine Rezeptaktion führt die von Ihnen angegebene Datentransformation aus. Eine Gruppe von Rezeptaktionen wird nacheinander ausgeführt, um den endgültigen Datensatz zu erstellen.

JSON

> Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Mitglied eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die JSON-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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"
        }
    }
}
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

YAML

Das Folgende zeigt ein BeispielRecipeAction, das als Teil eines Beispiels RecipeStep für ein DataBrew Rezept verwendet werden kann, wobei die YAML-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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'
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

AUSREISSER ENTFERNEN

Entfernt Datenpunkte, die auf der Grundlage der Einstellungen in den Parametern als Ausreißer klassifiziert werden.

Parameters

- `sourceColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `outlierStrategy`— Gibt den Ansatz an, der bei der Erkennung von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - `Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht.
  - `MODIFIED_Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die absolute Abweichung vom Median abweicht.
  - `IQR`— Identifiziert einen Wert als Ausreißer, wenn er über das erste und letzte Quartil der Spaltendaten hinausgeht. Der Interquartilsbereich (IQR) gibt an, wo sich die mittleren 50% der Datenpunkte befinden.
- `threshold`— Gibt den Schwellenwert an, der bei der Erkennung von Ausreißern verwendet werden soll. Der `sourceColumn` Wert wird als Ausreißer identifiziert, wenn der Wert, der mit dem berechnet wird, diese Zahl `outlierStrategy` überschreitet. Die Voreinstellung ist 3.
- `removeType`— Gibt an, wie die Daten entfernt werden sollen. Gültige Werte sind `DELETE_ROWS` und `CLEAR`.
- `trimValue`— Gibt an, ob alle oder einige Ausreißer entfernt werden sollen. Dieser boolesche Wert ist standardmäßig auf. `FALSE`
  - `FALSE`— Entfernt alle Ausreißer
  - `TRUE`— Entfernt Ausreißer, deren Rang außerhalb des in und angegebenen Perzentilschwellenwerts liegt. `minValue` `maxValue`
- `minValue`— Gibt den minimalen Perzentilwert für den Ausreißerbereich an. Der gültige Bereich liegt zwischen 0 und 100.
- `maxValue`— Gibt den maximalen Perzentilwert für den Ausreißerbereich an. Der gültige Bereich liegt zwischen 0 und 100.

Die folgenden Beispiele zeigen die Syntax für eine einzelne RecipeActionOperation. Ein Rezept enthält mindestens eine RecipeStepOperation, und ein Rezeptschritt enthält mindestens eine

Rezeptaktion. Eine Rezeptaktion führt die von Ihnen angegebene Datentransformation aus. Eine Gruppe von Rezeptaktionen wird nacheinander ausgeführt, um den endgültigen Datensatz zu erstellen.

JSON

Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Mitglied eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die JSON-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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"
        }
    }
}
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

YAML

Das Folgende zeigt ein Beispiel`RecipeAction`, das als Teil eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die YAML-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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'
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

AUSREISSER ERSETZEN

Aktualisiert die Datenpunktwerte, die als Ausreißer klassifiziert werden, auf der Grundlage der Einstellungen in den Parametern.

Parameters

- `sourceColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `outlierStrategy`— Gibt den Ansatz an, der bei der Erkennung von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - `Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht.
  - `MODIFIED_Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die absolute Abweichung vom Median abweicht.
  - `IQR`— Identifiziert einen Wert als Ausreißer, wenn er über das erste und letzte Quartil der Spaltendaten hinausgeht. Der Interquartilsbereich (IQR) gibt an, wo sich die mittleren 50% der Datenpunkte befinden.
- `threshold`— Gibt den Schwellenwert an, der bei der Erkennung von Ausreißern verwendet werden soll. Der `sourceColumn` Wert wird als Ausreißer identifiziert, wenn der Wert, der mit dem berechnet wird, diese Zahl `outlierStrategy` überschreitet. Die Voreinstellung ist 3.
- `replaceType`— Gibt die Methode an, die beim Ersetzen von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:

  - `WINSORIZE_VALUES`— Gibt an, dass das minimale und das maximale Perzentil zur Obergrenze der Werte verwendet werden.
  - `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`— Gibt den Typ der Modalfunktion an, die verwendet werden soll, wenn sie ist. `replaceType` `REPLACE_WITH_MODE` Zu den gültigen Werten gehören die folgenden: `MINMAX`, und`AVERAGE`.
- `minValue`— Gibt den minimalen Perzentilwert für den Ausreißerbereich an, der angewendet werden soll, wenn er verwendet wird. `trimValue` Der gültige Bereich liegt zwischen 0 und 100.
- `maxValue`— Gibt den maximalen Perzentilwert für den Ausreißerbereich an, der angewendet werden soll, wenn er verwendet wird. `trimValue` Der gültige Bereich liegt zwischen 0 und 100.
- `value`— Gibt den Wert an, der bei Verwendung eingefügt werden soll. `REPLACE_WITH_CUSTOM`
- `trimValue`— Gibt an, ob alle oder einige Ausreißer entfernt werden sollen. Dieser boolesche Wert ist auf `TRUE` when is,`REPLACE_WITH_NULL`, `replaceType` `REPLACE_WITH_MODE` or gesetzt. `WINSORIZE_VALUES` Für alle anderen ist er standardmäßig `FALSE` auf.
  - `FALSE`— Entfernt alle Ausreißer
  - `TRUE`— Entfernt Ausreißer, deren Rang außerhalb des in und angegebenen Schwellenwerts für die Perzentilobergrenze liegt. `minValue` `maxValue`

Die folgenden Beispiele zeigen die Syntax für einen einzelnen Vorgang. RecipeAction Ein Rezept enthält mindestens eine RecipeStepOperation, und ein Rezeptschritt enthält mindestens eine Rezeptaktion. Eine Rezeptaktion führt die von Ihnen angegebene Datentransformation aus. Eine Gruppe von Rezeptaktionen wird nacheinander ausgeführt, um den endgültigen Datensatz zu erstellen.

JSON

Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Mitglied eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die JSON-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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"
        }
    }
}
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

YAML

Das Folgende zeigt ein Beispiel`RecipeAction`, das als Teil eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die YAML-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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'
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

AUSREISSER MIT Z-SCORE NEU SKALIEREN

Gibt eine neue Spalte mit einem neu skalierten Ausreißerwert in jeder Zeile zurück, basierend auf den Einstellungen in den Parametern. Diese Aktion wendet auch eine Z-score Normalisierung auf linear skalierte Datenwerte an, sodass sie einen Mittelwert (μ) von 0 und eine Standardabweichung (μ) von 1 haben. Wir empfehlen diese Aktion für den Umgang mit Ausreißern.

Parameters

- `sourceColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `targetColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `outlierStrategy`— Gibt den Ansatz an, der bei der Erkennung von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - `Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht.
  - `MODIFIED_Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die absolute Abweichung vom Median abweicht.
  - `IQR`— Identifiziert einen Wert als Ausreißer, wenn er über das erste und letzte Quartil der Spaltendaten hinausgeht. Der Interquartilsbereich (IQR) gibt an, wo sich die mittleren 50% der Datenpunkte befinden.
- `threshold`— Der Schwellenwert, der bei der Erkennung von Ausreißern verwendet werden soll. Der `sourceColumn` Wert wird als Ausreißer identifiziert, wenn der Wert, der mit dem berechnet wird, diese Zahl `outlierStrategy` überschreitet. Die Voreinstellung ist 3.

Die folgenden Beispiele zeigen die Syntax für eine einzelne RecipeActionOperation. Ein Rezept enthält mindestens eine RecipeStepOperation, und ein Rezeptschritt enthält mindestens eine Rezeptaktion. Eine Rezeptaktion führt die von Ihnen angegebene Datentransformation aus. Eine Gruppe von Rezeptaktionen wird nacheinander ausgeführt, um den endgültigen Datensatz zu erstellen.

JSON

Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Mitglied eines Beispiels `RecipeStep` für eine DataBrew Rezeptoperation unter Verwendung der JSON-Syntax verwendet werden kann. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in JSON

```
{
    "Action": {
        "Operation": "RESCALE_OUTLIERS_WITH_Z_SCORE",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "Z_SCORE",
            "threshold": "3"
        }
    }
}
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

YAML

Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Teil eines Beispiels `RecipeStep` für eine DataBrew Recipe-Operation unter Verwendung der YAML-Syntax verwendet werden kann. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in YAML

```
- Action:
```

```
Operation: REMOVE_OUTLIERS
Parameters:
  sourceColumn: name-of-existing-column
  targetColumn: name-of-new-column
  outlierStrategy: Z_SCORE
  threshold: '3'
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

AUSREISSER MIT SCHRÄGLAGE NEU SKALIEREN

Gibt eine neue Spalte mit einem neu skalierten Ausreißerwert in jeder Zeile zurück, basierend auf den Einstellungen in den Parametern. Diese Aktion dient dazu, die Verteilungsschiefe zu verringern, indem die angegebene Protokoll- oder Stammtransformation angewendet wird. Wir empfehlen diese Aktion für den Umgang mit verzerrten Daten.

Parameters

- `sourceColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `targetColumn`— Gibt den Namen einer vorhandenen numerischen Spalte an, die Ausreißer enthalten könnte.
- `outlierStrategy`— Gibt den Ansatz an, der bei der Erkennung von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - `Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die Standardabweichung vom Mittelwert abweicht.
  - `MODIFIED_Z_SCORE`— Identifiziert einen Wert als Ausreißer, wenn er um mehr als den Schwellenwert für die absolute Abweichung vom Median abweicht.
  - `IQR`— Identifiziert einen Wert als Ausreißer, wenn er über das erste und letzte Quartil der Spaltendaten hinausgeht. Der Interquartilsbereich (IQR) gibt an, wo sich die mittleren 50% der Datenpunkte befinden.
- `threshold`— Gibt den Schwellenwert an, der bei der Erkennung von Ausreißern verwendet werden soll. Der `sourceColumn` Wert wird als Ausreißer identifiziert, wenn der Wert, der mit dem berechnet wird, diese Zahl `outlierStrategy` überschreitet. Die Voreinstellung ist 3.

- `skewFunction`— Gibt die Methode an, die beim Ersetzen von Ausreißern verwendet werden soll. Gültige Werte sind z. B. die Folgenden:
  - LOG — Wendet eine starke Transformation an, um positive und negative Verzerrungen zu reduzieren. Dies ist ein natürlicher Logarithmus (2.718281828).
  - WURZEL (mit`value = 3`) — Wendet eine ziemlich starke Transformation an, um positive und negative Verzerrungen zu reduzieren. (Kubikwurzel)
  - WURZEL (mit`value = 2`) — Wendet eine moderate Transformation an, um nur die positive Verzerrung zu reduzieren. (Quadratwurzel)
  - SQUARE — Wendet eine moderate Transformation an, um negative Verzerrungen zu reduzieren. (Quadratisch)
  - Benutzerdefinierte Transformation — Wendet die angegebene `LOG` oder die `ROOT` Transformation unter Verwendung der im `value` Parameter angegebenen benutzerdefinierten Zahl an.
- `value`— Gibt den Wert an, der für die benutzerdefinierte Transformation verwendet werden soll. Wenn LOG `skewFunction` ist, stellt dieser Wert die Basis des Protokolls dar. Wenn ROOT `skewFunction` ist, steht dieser Wert für die Potenz der Wurzel.

Die folgenden Beispiele zeigen die Syntax für eine einzelne RecipeActionOperation. Ein Rezept enthält mindestens eine RecipeStepOperation, und ein Rezeptschritt enthält mindestens eine Rezeptaktion. Eine Rezeptaktion führt die von Ihnen angegebene Datentransformation aus. Eine Gruppe von Rezeptaktionen wird nacheinander ausgeführt, um den endgültigen Datensatz zu erstellen.

JSON

Im Folgenden wird ein Beispiel gezeigt`RecipeAction`, das als Mitglied eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die JSON-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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"
        }
    }
}
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

YAML

Das Folgende zeigt ein Beispiel`RecipeAction`, das als Teil eines Beispiels `RecipeStep` für ein DataBrew Rezept verwendet werden kann, wobei die YAML-Syntax verwendet wird. Syntaxbeispiele mit einer Liste von Rezeptaktionen finden Sie unterDefinition einer Rezeptstruktur.

Example Beispiel in 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'
```

Weitere Informationen zur Verwendung dieser Rezeptaktion in einer API-Operation finden Sie unter CreateRecipeoder UpdateRecipe. Sie können diese und andere API-Operationen in Ihrem eigenen Code verwenden.

Rezeptschritte für die Spaltenstruktur

Verwenden Sie diese Rezeptschritte für die Spaltenstruktur, um die Spaltenstruktur Ihrer Daten zu ändern.

Themen

- BOOLESCHE_OPERATION
- CASE_OPERATION
- FLAG_COLUMN_FROM_NULL
- FLAG_COLUMN_FROM_PATTERN
- MERGE
- SPLIT_COLUMN_BETWEEN_DELIMITER
- SPLIT_COLUMN_BETWEEN_POSITIONS
- SPLIT_COLUMN_FROM_END
- SPLIT_COLUMN_FROM_START
- SPLIT_COLUMN_MULTIPLE_DELIMITER
- SPLIT_COLUMN_SINGLE_DELIMITER
- SPLIT_COLUMN_WITH_INTERVALS

BOOLESCHE_OPERATION

Erstellt eine neue Spalte, die auf dem Ergebnis der logischen Bedingung IF basiert. Gibt den Wert wahr zurück, wenn der boolesche Ausdruck wahr ist, den Wert falsch, wenn der boolesche Ausdruck falsch ist, oder gibt einen benutzerdefinierten Wert zurück.

Parameters

- `trueValueExpression`— Ergebnis, wenn die Bedingung erfüllt ist.
- `falseValueExpression`— Ergebnis, wenn die Bedingung nicht erfüllt ist.
- `valueExpression`— Boolesche Bedingung.
- `withExpressions`— Konfiguration für aggregierte Ergebnisse.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Sie können konstante Werte, Spaltenverweise und aggregierte Ergebnisse in TrueValueExpression, False ValueExpression und ValueExpression verwenden.

Example Beispiel: Konstante Werte

Werte, die unverändert bleiben, wie eine Zahl oder ein Satz.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Beispiel: Spaltenverweise

Werte, die Spalten im Datensatz sind.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Beispiel: Ergebnisse aggregieren

Werte, die durch Aggregatfunktionen berechnet werden. Eine Aggregatfunktion führt eine Berechnung für eine Spalte durch und gibt einen einzelnen Wert zurück.

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

Benutzer müssen den JSON-Code durch Escape-Zeichen in eine Zeichenfolge konvertieren.

Beachten Sie, dass die Parameternamen in true ValueExpressionValueExpression, false und ValueExpression mit den Namen in withExpressions übereinstimmen müssen. Um die Aggregatergebnisse einiger Spalten zu verwenden, müssen Sie Parameter für sie erstellen und die Aggregatfunktionen bereitstellen.

Example Beispiel:

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Beispiel: and/or

Sie können und und oder verwenden, um mehrere Bedingungen zu kombinieren.

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

Gültige Aggregatfunktionen

Die folgende Tabelle zeigt alle gültigen Aggregatfunktionen, die in einer booleschen Operation verwendet werden können.

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
Numerischer Wert	Summe	`:sum.column.1`	`[ { "name": "sum.column.1",`	Gibt die Summe von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
			"value": "sum(`column.1`)", "type": "aggregate" }]	
	Mean	`:mean.column.1`	[{ "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" }]	Gibt den Mittelwert von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Mittlere absolute Abweichung	`:mittlere absolute Abweichung. Spalte 1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Gibt die mittlere absolute Abweichung von `column.1`
	Median	`:median. column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Gibt den Median von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Produkt	`:product .column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Gibt das Produkt von `column.1`
	Standardabweichung	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Gibt die Standardabweichung von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Varianz	`:varianc e.column.1`	`[ { "name": "variance .column.1 ", "value": "variance (`column. 1`)", "type": "aggregat e" } ]`	Gibt die Varianz von `column.1`
	Standardfehler des Mittelwerts	`:Standar derrorOfM ean.Column.1`	`[ { "name": "standard errorofme an.column .1", "value": "standard _error_of _mean(`co lumn.1`)", "type": "aggregat e" } ]`	Gibt den Standardfehler des Mittelwerts von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Schiefe	`:Schiefe .Spalte.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Gibt die Schiefe von zurück `column.1`
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Gibt die Kurtosis von zurück `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
Datetime/ Numeric/Text	Anzahl	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Gibt die Gesamtzahl der Zeilen zurück in `column.1`
	Count (Distinct)	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Gibt die Gesamtzahl der unterschiedlichen Zeilen zurück in `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Min	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Gibt den Minimalwert von `column.1`
	Max	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Gibt den Maximalwert von `column.1`

Gültige Bedingungen in einem ValueExpression

Die folgende Tabelle zeigt die unterstützten Bedingungen und die Werteausdrücke, die Sie verwenden können.

Spaltentyp	Bedingung	ValueExpression	Description
Zeichenfolge	Enthält	enthält (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte Text enthält
	Enthält keinen	! enthält (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte keinen Text enthält
	Entspricht	entspricht (`Spalte`, 'Muster')	Bedingung, um zu testen, ob der Wert in der Spalte dem Muster entspricht
	Stimmt nicht überein	! entspricht (`Spalte`, 'Muster')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit dem Muster übereinstimmt
	Beginnt mit	StartsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte mit Text beginnt
	Beginnt nicht mit	! StartsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit Text beginnt
	Endet mit	EndsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte mit Text endet

Spaltentyp	Bedingung	ValueExpression	Description
	Endet nicht mit	! EndsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit Text endet
Numerischer Wert	Kleiner als	`Spalte` < Zahl	Bedingung, um zu testen, ob der Wert in der Spalte kleiner als Zahl ist
	Kleiner als oder gleich	`Spalte` <= Zahl	Bedingung, um zu testen, ob der Wert in der Spalte kleiner oder gleich einer Zahl ist
	Größer als	`Spalte` > Zahl	Bedingung, um zu testen, ob der Wert in der Spalte größer als Zahl ist
	Größer als oder gleich	`Spalte` >= Zahl	Bedingung, um zu testen, ob der Wert in der Spalte größer oder gleich einer Zahl ist
	Ist zwischen	isBetween (`column` , minNumber, maxNumber)	Bedingung, um zu testen, ob der Wert in der Spalte zwischen minNumber und maxNumber liegt

Spaltentyp	Bedingung	ValueExpression	Description
	Liegt nicht zwischen	! isBetween (`Spalte` , MinNumber, MaxNumber)	Bedingung, um zu testen, ob der Wert in der Spalte nicht zwischen minNumber und maxNumber liegt
Boolesch	Ist wahr	`column` = WAHR	Bedingung, um zu testen, ob der Wert in der Spalte boolean TRUE ist
	Ist falsch	`column` = FALSCH	Bedingung, um zu testen, ob der Wert in der Spalte ein boolescher Wert ist FALSE
Date/Timestamp	Früher als	`column` < 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte vor dem Datum liegt
	Früher als oder gleich	`column` <= 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte vor oder gleich dem Datum liegt
	Später als	`column` > 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte nach dem Datum liegt

Spaltentyp	Bedingung	ValueExpression	Description
	Später als oder gleich	`column` >= 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte später als oder gleich dem Datum liegt
String/Numeric/Date/ Timestamp	Ist genau	`column` = 'Wert'	Bedingung, um zu testen, ob der Wert in der Spalte exakt dem Wert entspricht
	Ist nicht	`Spalte`! = 'Wert'	Bedingung, um zu testen, ob der Wert in der Spalte kein Wert ist
	Fehlt	fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte fehlt
	Fehlt nicht	! fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte nicht fehlt
	Ist gültig	isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)

Spaltentyp	Bedingung	ValueExpression	Description
	Ist nicht gültig	! isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte nicht gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)
Verschachtelt	Fehlt	fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte fehlt
	Fehlt nicht	! fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte nicht fehlt
	Ist gültig	isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)
	Ist nicht gültig	! isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte nicht gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)

CASE_OPERATION

Erstellen Sie eine neue Spalte, die auf dem Ergebnis der logischen Bedingung CASE basiert. Die Falloperation durchläuft die Fallbedingungen und gibt einen Wert zurück, wenn die erste Bedingung erfüllt ist. Sobald eine Bedingung erfüllt ist, beendet die Operation den Lesevorgang und gibt das Ergebnis zurück. Wenn keine Bedingungen erfüllt sind, wird der Standardwert zurückgegeben.

Parameters

- `valueExpression`— Bedingungen.
- `withExpressions`— Konfiguration für aggregierte Ergebnisse.
- `targetColumn`— Name für die neu erstellte Spalte.

Example Beispiel

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

Gültige Aggregatfunktionen

Die folgende Tabelle zeigt alle gültigen Aggregatfunktionen, die in einer Falloperation verwendet werden können.

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
Numerischer Wert	Summe	`:sum.column.1`	`[ { "name":`	Gibt die Summe von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
			"sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" }]	
	Mean	`:mean.column.1`	[{ "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" }]	Gibt den Mittelwert von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Mittlere absolute Abweichung	`:mittlere absolute Abweichung. Spalte 1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Gibt die mittlere absolute Abweichung von `column.1`
	Median	`:median. column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Gibt den Median von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Produkt	`:product .column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Gibt das Produkt von `column.1`
	Standardabweichung	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Gibt die Standardabweichung von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Varianz	`:varianc e.column.1`	`[ { "name": "variance .column.1 ", "value": "variance (`column. 1`)", "type": "aggregat e" } ]`	Gibt die Varianz von `column.1`
	Standardfehler des Mittelwerts	`:Standar derrorOfM ean.Column.1`	`[ { "name": "standard errorofme an.column .1", "value": "standard _error_of _mean(`co lumn.1`)", "type": "aggregat e" } ]`	Gibt den Standardfehler des Mittelwerts von `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Schiefe	`:Schiefe .Spalte.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Gibt die Schiefe von zurück `column.1`
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Gibt die Kurtosis von zurück `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
Datetime/ Numeric/Text	Anzahl	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Gibt die Gesamtzahl der Zeilen zurück in `column.1`
	Count (Distinct)	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Gibt die Gesamtzahl der unterschiedlichen Zeilen zurück in `column.1`

Typ der Spalte	Bedingung	ValueExpression	Mit Ausdrücken	Rückgabewert
	Min	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Gibt den Minimalwert von `column.1`
	Max	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Gibt den Maximalwert von `column.1`

Gültige Bedingungen in einem ValueExpression

Die folgende Tabelle zeigt die unterstützten Bedingungen und die Werteausdrücke, die Sie verwenden können.

Spaltentyp	Bedingung	ValueExpression	Description
Zeichenfolge	Enthält	enthält (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte Text enthält
	Enthält keinen	! enthält (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte keinen Text enthält
	Entspricht	entspricht (`Spalte`, 'Muster')	Bedingung, um zu testen, ob der Wert in der Spalte dem Muster entspricht
	Stimmt nicht überein	! entspricht (`Spalte`, 'Muster')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit dem Muster übereinstimmt
	Beginnt mit	StartsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte mit Text beginnt
	Beginnt nicht mit	! StartsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit Text beginnt
	Endet mit	EndsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte mit Text endet

Spaltentyp	Bedingung	ValueExpression	Description
	Endet nicht mit	! EndsWith (`Spalte`, 'Text')	Bedingung, um zu testen, ob der Wert in der Spalte nicht mit Text endet
Numerischer Wert	Kleiner als	`Spalte` < Zahl	Bedingung, um zu testen, ob der Wert in der Spalte kleiner als Zahl ist
	Kleiner als oder gleich	`Spalte` <= Zahl	Bedingung, um zu testen, ob der Wert in der Spalte kleiner oder gleich einer Zahl ist
	Größer als	`Spalte` > Zahl	Bedingung, um zu testen, ob der Wert in der Spalte größer als Zahl ist
	Größer als oder gleich	`Spalte` >= Zahl	Bedingung, um zu testen, ob der Wert in der Spalte größer oder gleich einer Zahl ist
	Ist zwischen	isBetween (`column` , minNumber, maxNumber)	Bedingung, um zu testen, ob der Wert in der Spalte zwischen minNumber und maxNumber liegt

Spaltentyp	Bedingung	ValueExpression	Description
	Liegt nicht zwischen	! isBetween (`Spalte` , MinNumber, MaxNumber)	Bedingung, um zu testen, ob der Wert in der Spalte nicht zwischen minNumber und maxNumber liegt
Boolesch	Ist wahr	`column` = WAHR	Bedingung, um zu testen, ob der Wert in der Spalte boolean TRUE ist
	Ist falsch	`column` = FALSCH	Bedingung, um zu testen, ob der Wert in der Spalte ein boolescher Wert ist FALSE
Date/Timestamp	Früher als	`column` < 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte vor dem Datum liegt
	Früher als oder gleich	`column` <= 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte vor oder gleich dem Datum liegt
	Später als	`column` > 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte nach dem Datum liegt

Spaltentyp	Bedingung	ValueExpression	Description
	Später als oder gleich	`column` >= 'Datum'	Bedingung, um zu testen, ob der Wert in der Spalte später als oder gleich dem Datum liegt
String/Numeric/Date/Timestamp	Ist genau	`column` = 'Wert'	Bedingung, um zu testen, ob der Wert in der Spalte exakt dem Wert entspricht
	Ist nicht	`Spalte`! = 'Wert'	Bedingung, um zu testen, ob der Wert in der Spalte kein Wert ist
	Fehlt	fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte fehlt
	Fehlt nicht	! fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte nicht fehlt
	Ist gültig	isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)

Spaltentyp	Bedingung	ValueExpression	Description
	Ist nicht gültig	! isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte nicht gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)
Verschachtelt	Fehlt	fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte fehlt
	Fehlt nicht	! fehlt (`Spalte`)	Bedingung, um zu testen, ob der Wert in der Spalte nicht fehlt
	Ist gültig	isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)
	Ist nicht gültig	! isValid (`Spalte`, Datentyp)	Bedingung, um zu testen, ob der Wert in der Spalte nicht gültig ist (der Wert ist vom Datentyp oder er kann in einen Datentyp konvertiert werden)

FLAG_COLUMN_FROM_NULL

Erstellt eine neue Spalte, die auf dem Vorhandensein von Nullwerten in einer vorhandenen Spalte basiert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn`— Der Name einer neuen Spalte, die erstellt werden soll.
- `flagType`— Ein Wert, auf den gesetzt werden muss`Null values`.
- `trueString`— Ein Wert für die neue Spalte, wenn in der Quelle ein Nullwert gefunden wird. Wenn kein Wert angegeben wird, lautet der Standardwert `True`.
- `falseString`— Ein Wert für die neue Spalte, wenn in der Quelle ein Wert gefunden wird, der nicht Null ist. Wenn kein Wert angegeben wird, lautet der Standardwert `False`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

Erstellt eine neue Spalte, die auf dem Vorhandensein eines benutzerdefinierten Musters in einer vorhandenen Spalte basiert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn`— Der Name einer neuen Spalte, die erstellt werden soll.
- `flagType`— Ein Wert, auf den gesetzt werden muss`Pattern`.

- `pattern`— Ein regulärer Ausdruck, der das auszuwertende Muster angibt.
- `trueString`— Ein Wert für die neue Spalte, wenn in der Quelle ein Nullwert gefunden wird. Wenn kein Wert angegeben wird, lautet der Standardwert `True`.
- `falseString`— Ein Wert für die neue Spalte, wenn in der Quelle ein Wert gefunden wird, der nicht Null ist. Wenn kein Wert angegeben wird, lautet der Standardwert `False`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
    }
}
```

MERGE

Führt zwei oder mehr Spalten zu einer neuen Spalte zusammen.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste mit einer oder mehreren Spalten darstellt, die zusammengeführt werden sollen.
- `delimiter`— Ein optionales Trennzeichen zwischen den Werten, das in der Zielspalte erscheinen soll.
- `targetColumn`— Der Name der zusammengeführten Spalte, die erstellt werden soll.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_DELIMITER

Teilt eine Spalte entsprechend einem Anfangs- und Endtrennzeichen in drei neue Spalten auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `patternOption1`— Eine JSON-encoded Zeichenfolge, die ein oder mehrere Zeichen darstellt, die das erste Trennzeichen angeben.
- `patternOption2`— Eine JSON-encoded Zeichenfolge, die ein oder mehrere Zeichen darstellt, die das zweite Trennzeichen angeben.
- `pattern`— Ein oder mehrere Zeichen, die bei der Aufteilung der Daten als Trennzeichen verwendet werden sollen.
- `includeInSplit`— Falls wahr, wird das Muster in die neue Spalte aufgenommen; andernfalls wird das Muster verworfen.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_DELIMITER",
        "Parameters": {
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_POSITIONS

Teilt eine Spalte entsprechend den von Ihnen angegebenen Offsets in drei neue Spalten auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `startPosition`— Die Position des Zeichens, an der die Teilung beginnen soll.
- `endPosition`— Die Position des Zeichens, an der die Teilung enden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
        }
    }
}
```

SPLIT_COLUMN_FROM_END

Teilt eine Spalte mit einem Abstand vom Ende der Zeichenfolge in zwei neue Spalten auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `position`— Die Position des Zeichens vom rechten Ende der Zeichenfolge aus, an der die Aufteilung erfolgen soll.

Example Beispiel

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

```
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
}
```

SPLIT_COLUMN_FROM_START

Teilt eine Spalte mit einem Abstand vom Anfang der Zeichenfolge in zwei neue Spalten auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `position`— Die Position des Zeichens vom linken Ende der Zeichenfolge aus, an der die Aufteilung erfolgen soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

SPLIT_COLUMN_MULTIPLE_DELIMITER

Teilt eine Spalte nach mehreren Trennzeichen auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `patternOptions`— Eine JSON-encoded Zeichenfolge, die ein oder mehrere Muster darstellt, die die Teilungskriterien bestimmen.

- `pattern`— Ein oder mehrere Zeichen, die bei der Aufteilung der Daten als Trennzeichen verwendet werden sollen.
- `limit`— Wie viele Splits durchgeführt werden sollen. Das Minimum ist 1; das Maximum ist 20.
- `includeInSplit`— Falls wahr, wird das Muster in die neue Spalte aufgenommen; andernfalls wird das Muster verworfen.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
}
```

SPLIT_COLUMN_SINGLE_DELIMITER

Teilt eine Spalte entsprechend einem bestimmten Trennzeichen in eine oder mehrere neue Spalten auf.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Ein oder mehrere Zeichen, die bei der Aufteilung der Daten als Trennzeichen verwendet werden.
- `limit`— Wie viele Splits durchgeführt werden sollen. Das Minimum ist 1; das Maximum ist 20.
- `includeInSplit`— Falls wahr, wird das Muster in die neue Spalte aufgenommen; andernfalls wird das Muster verworfen.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_SINGLE_DELIMITER",
        "Parameters": {
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVALS

Teilt eine Spalte in Intervallen von n Zeichen, wobei Sie n angeben.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `startPosition`— Die Zeichenposition, an der die Teilung beginnen soll.
- `interval`— Die Anzahl der Zeichen, die vor dem nächsten Split übersprungen werden sollen.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

Rezeptschritte zur Spaltenformatierung

Verwenden Sie die Anleitungen zur Spaltenformatierung, um das Format der Daten in Ihren Spalten zu ändern.

Themen

- NUMBER_FORMAT
- TELEFONNUMMER_FORMATIEREN

NUMBER_FORMAT

Gibt eine Spalte zurück, in der ein numerischer Wert in eine formatierte Zeichenfolge umgewandelt wird.

Parameters

- `sourceColumn` – Zeichenfolge. Der Name einer vorhandenen Spalte.
- `decimalPlaces`— Ganzzahl. Der Wert der Anzahl der Ziffern nach dem Dezimaltrennzeichen.
- `numericDecimalSeparator` – Zeichenfolge. Einer der folgenden Werte gibt das Dezimaltrennzeichen an:
  - "."
  - ","
- `numericThousandSeparator` – Zeichenfolge. Einer der folgenden Werte gibt das Tausendertrennzeichen an:
  - null. Zeigt an, dass ein Tausendertrennzeichen nicht aktiviert ist.
  - ","
  - " "
  - "."
  - "\\"
- `numericAbbreviatedUnit` – Zeichenfolge. Einer der folgenden Werte gibt die Abkürzungseinheit an:
  - null. Zeigt an, dass eine Abkürzungseinheit nicht aktiviert ist.
  - „TAUSEND“
  - „MILLIONEN“
  - „MILLIARDE“
  - „BILLION“
- `numericUnitAbbreviation` – Zeichenfolge. Einer der folgenden Werte oder ein beliebiger benutzerdefinierter Wert, der die Abkürzung der Einheit angibt:
  - null. Zeigt an, dass die Abkürzung für Einheiten nicht aktiviert ist.

Einheit der Abkürzung	Optionen
Tausende	K, k, M, tausend, benutzerdefiniert
Millionen	M, m, MM, Millionen, benutzerdefiniert
Milliarde	B, bn, Milliarde, benutzerdefiniert
Billion	T, zehn, Billionen, benutzerdefiniert

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
            "numericUnitAbbreviation": "K"
        }
    }
}
```

TELEFONNUMMER_FORMATIEREN

Gibt eine Spalte zurück, in der eine Telefonnummernzeichenfolge in einen formatierten Wert umgewandelt wird.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `phoneNumberFormat` – Das Format, in das die Telefonnummer konvertiert werden soll. Wenn kein Format angegeben ist, wird das Standardformat `E.164` verwendet, ein international anerkanntes Standardformat für Telefonnummern. Gültige Werte sind z. B. die Folgenden:

  - `E164`(lassen Sie den Punkt danach weg) `E`
- `defaultRegion` – Ein gültiger Regionscode, der aus zwei oder drei Großbuchstaben besteht und die Region für die Telefonnummer angibt, wenn in der Nummer selbst keine Landesvorwahl enthalten ist. Es kann höchstens `defaultRegion` oder `defaultRegionColumn` angegeben werden.
- `defaultRegionColumn`— Der Name einer Spalte des erweiterten Datentyps`Country`. Der Regionscode aus der angegebenen Spalte wird verwendet, um die Landesvorwahl für die Telefonnummer zu ermitteln, wenn in der Nummer selbst keine Landesvorwahl vorhanden ist. Es kann höchstens `defaultRegion` oder `defaultRegionColumn` angegeben werden.

Hinweise

- Eingaben, die nicht mit einer gültigen Telefonnummer formatiert werden können, bleiben unverändert.
- Wenn keine Standardregion angegeben ist und eine Telefonnummer nicht mit einem Pluszeichen (+) und einer Landesvorwahl beginnt, ist die Telefonnummer nicht formatiert.

Example

Beispiel: Die Standardregion wurde korrigiert

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

Beispiel: Standardoption für die Spalte „Region“

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
```

```
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

Rezeptschritte für die Datenstruktur

Verwenden Sie diese Rezeptschritte, um Daten aus verschiedenen Perspektiven tabellarisch darzustellen und zusammenzufassen oder um erweiterte Funktionen auszuführen.

Themen

- NEST_TO_ARRAY
- NEST_TO_MAP
- NEST_TO_STRUCT
- UNNEST_ARRAY
- UNNEST_MAP
- UNNEST_STRUCT
- UNNEST_STRUCT_N
- GROUP_BY
- JOIN
- PIVOT
- SCALE
- TRANSPONIEREN
- UNION
- UNPIVOT

NEST_TO_ARRAY

Konvertiert vom Benutzer ausgewählte Spalten in Array-Werte. Die Reihenfolge der ausgewählten Spalten wird bei der Erstellung des resultierenden Arrays beibehalten. Die verschiedenen Spaltendatentypen werden in einen gemeinsamen Typ umgewandelt, der die Datentypen aller Spalten unterstützt.

Parameters

- `sourceColumns`— Liste der Quellspalten.
- `targetColumn`— Der Name der Zielspalte.
- `removeSourceColumns`— Enthält den Wert `true` oder gibt `false` an, ob der Benutzer die ausgewählten Quellspalten entfernen möchte oder nicht.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_MAP

Konvertiert vom Benutzer ausgewählte Spalten in Schlüssel-Wert-Paare, wobei jeweils ein Schlüssel den Spaltennamen und ein Wert den Zeilenwert darstellt. Die Reihenfolge der ausgewählten Spalte wird bei der Erstellung der resultierenden Map nicht beibehalten. Die verschiedenen Spaltendatentypen werden in einen gemeinsamen Typ umgewandelt, der die Datentypen aller Spalten unterstützt.

Parameters

- `sourceColumns`— Liste der Quellspalten.
- `targetColumn`— Der Name der Zielspalte.
- `removeSourceColumns`— Enthält den Wert `true` oder gibt `false` an, ob der Benutzer die ausgewählten Quellspalten entfernen möchte oder nicht.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_STRUCT

Konvertiert vom Benutzer ausgewählte Spalten in Schlüssel-Wert-Paare, wobei jeweils ein Schlüssel den Spaltennamen und ein Wert den Zeilenwert darstellt. Die Reihenfolge der ausgewählten Spalten und der Datentyp jeder Spalte werden in der resultierenden Struktur beibehalten.

Parameters

- `sourceColumns`— Liste der Quellspalten.
- `targetColumn`— Der Name der Zielspalte.
- `removeSourceColumns`— Enthält den Wert `true` oder gibt `false` an, ob der Benutzer die ausgewählten Quellspalten entfernen möchte oder nicht.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

UNNEST_ARRAY

Löscht eine Spalte vom Typ `array` in eine neue Spalte. Wenn das Array mehr als einen Wert enthält, wird eine Zeile generiert, die jedem Element entspricht. Diese Funktion entfernt nur eine Ebene einer Array-Spalte.

Parameters

- `sourceColumn`— Der Name einer vorhandenen Spalte. Diese Spalte muss vom `struct` Typ sein.
- `targetColumn`— Name der Zielspalte, die generiert wird.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_MAP

Löscht eine Spalte vom Typ `map` und generiert eine Spalte für den Schlüssel und den Wert. Wenn es mehr als ein Schlüssel-Wert-Paar gibt, würde eine Zeile generiert, die jedem Schlüsselwert entspricht. Diese Funktion entfernt nur eine Ebene einer Kartenspalte.

Parameters

- `sourceColumn`— Der Name einer vorhandenen Spalte. Diese Spalte muss vom `struct` Typ sein.
- `removeSourceColumn`— Falls`true`, wird die Quellspalte gelöscht, nachdem die Funktion abgeschlossen ist.
- `targetColumn`— Falls angegeben, beginnt jede der generierten Spalten mit diesem Präfix.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

Löscht eine Spalte vom Typ `struct` und generiert eine Spalte für jeden der in der Struktur vorhandenen Schlüssel. Diese Funktion entfernt nur die erste Strukturebene.

Parameters

- `sourceColumn`— Der Name einer vorhandenen Spalte. Diese Spalte muss vom Typ Structure sein.
- `removeSourceColumn`— Falls`true`, wird die Quellspalte gelöscht, nachdem die Funktion abgeschlossen ist.
- `targetColumn`— Falls angegeben, beginnt jede der generierten Spalten mit diesem Präfix.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

Erstellt eine neue Spalte für jedes Feld einer ausgewählten Spalte vom Typ. `struct`

Zum Beispiel mit der folgenden Struktur:

```
user {
   name: “Ammy”
   address: {
      state: "CA",
      zipcode: 12345
   }
}
```

Diese Funktion erstellt 3 Spalten:

user.name	benutzer.adresse.state	Benutzer.Adresse.Postleitzahl
Ammy	CA	12345

Parameters

- `sourceColumns`— Liste der Quellspalten.
- `regexColumnSelector`— Ein regulärer Ausdruck zur Auswahl der Spalten, deren Verschachtelung aufgehoben werden soll.
- `removeSourceColumn`— Ein boolescher Wert. Falls wahr, entfernen Sie die Quellspalte; andernfalls behalten Sie sie bei.
- `unnestLevel`— Die Anzahl der Ebenen, deren Verschachtelung aufgehoben werden soll.
- `delimiter`— Das Trennzeichen wird im Namen der neu erstellten Spalte verwendet, um die verschiedenen Ebenen der Struktur voneinander zu trennen. Beispiel: Wenn das Trennzeichen „/“ ist, hat der Spaltenname diese Form: „/state“. user/address
- `conditionExpressions`— Bedingungsausdrücke.

Example Beispiel

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

GROUP_BY

Fasst die Daten zusammen, indem Zeilen nach einer oder mehreren Spalten gruppiert werden und dann auf jede Gruppe eine Aggregationsfunktion angewendet wird.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spalten darstellt, die die Grundlage jeder Gruppe bilden.
- `groupByAggFunctions`— Eine JSON-encoded Zeichenfolge, die eine Liste der anzuwendenden Aggregationsfunktionen darstellt. (Wenn Sie keine Aggregation wünschen, geben Sie an`UNAGGREGATED`.)
- `useNewDataFrame`— Falls wahr, werden die Ergebnisse von GROUP_BY in der Projektsitzung verfügbar gemacht und ersetzen den aktuellen Inhalt.

Example Beispiel

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
```

```
      }
    }
  }
]
```

JOIN

Führt eine Verbindungsoperation für zwei Datensätze durch.

Parameters

- `joinKeys`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spalten aus jedem Datensatz darstellt, die als Join-Schlüssel dienen sollen.
- `joinType`— Die Art der auszuführenden Verknüpfung. Muss einer der folgenden sein: `INNER_JOIN` `LEFT_JOIN` | `RIGHT_JOIN` | `OUTER_JOIN` | `LEFT_EXCLUDING_JOIN` | `RIGHT_EXCLUDING_JOIN` | `OUTER_EXCLUDING_JOIN`
- `leftColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spalten aus der aktuellen aktiven Datenmenge darstellt.
- `rightColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spalten aus einem anderen (sekundären) Datensatz darstellt, die mit dem aktuellen Datensatz verknüpft werden sollen.
- `secondInputLocation`— Eine Amazon S3 S3-URL, die in die Datendatei für den sekundären Datensatz aufgelöst wird.
- `secondaryDatasetName`— Der Name des sekundären Datensatzes.

Example Beispiel

```
{
    "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

Konvertiert alle Zeilenwerte in einer ausgewählten Spalte in einzelne Spalten mit Werten.

Parameters

- `sourceColumn`— Der Name einer vorhandenen Spalte. Die Spalte kann maximal 10 verschiedene Werte haben.
- `valueColumn`— Der Name einer vorhandenen Spalte. Die Spalte kann maximal 10 verschiedene Werte haben.
- `aggregateFunction`— Der Name einer Aggregationsfunktion. Wenn Sie keine Aggregation wünschen, verwenden Sie das Schlüsselwort. `COLLECT_LIST`

Example Beispiel

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
            "valueColumn": "all_votes"
        }
    }
```

```
}
```

SCALE

Skaliert oder normalisiert den Datenbereich in einer numerischen Spalte.

Parameters

- `sourceColumn`— Der Name einer vorhandenen Spalte.
- `strategy`— Die Operation, die auf die Spaltenwerte angewendet werden soll:
  - `MIN_MAX`— Skaliert die Werte auf einen Bereich von [0,1] neu.
  - `SCALE_BETWEEN`— Skaliert die Werte in einen Bereich von zwei angegebenen Werten neu.
  - `MEAN_NORMALIZATION`— Skaliert die Daten neu, sodass sie einen Mittelwert (μ) von 0 und eine Standardabweichung (σ) von 1 innerhalb eines Bereichs von [-1, 1] haben.
  - `Z_SCORE`— Skaliert Datenwerte linear, sodass sie einen Mittelwert (μ) von 0 und eine Standardabweichung (μ) von 1 haben. Am besten für den Umgang mit Ausreißern geeignet.
- `targetColumn`— Der Name einer Spalte, die die Ergebnisse enthalten soll.

Example Beispiel

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

TRANSPONIEREN

Konvertiert alle ausgewählten Zeilen in Spalten und Spalten in Zeilen.

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`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spalten darstellt, deren Zeilen in Spaltennamen umgewandelt werden.
- `valueColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste mit einer oder mehreren Spalten darstellt, die in Zeilen umgewandelt werden sollen.
- `aggregateFunction`— Der Name einer Aggregationsfunktion. Wenn Sie keine Aggregation wünschen, verwenden Sie das Schlüsselwort. `COLLECT_LIST`
- `newColumn`— Die Spalte, die transponierte Spalten als Werte enthält.

Example Beispiel

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
            "newColumn": "Student"
        }
    }

}
```

UNION

Kombiniert die Zeilen aus zwei oder mehr Datensätzen zu einem einzigen Ergebnis.

Parameters

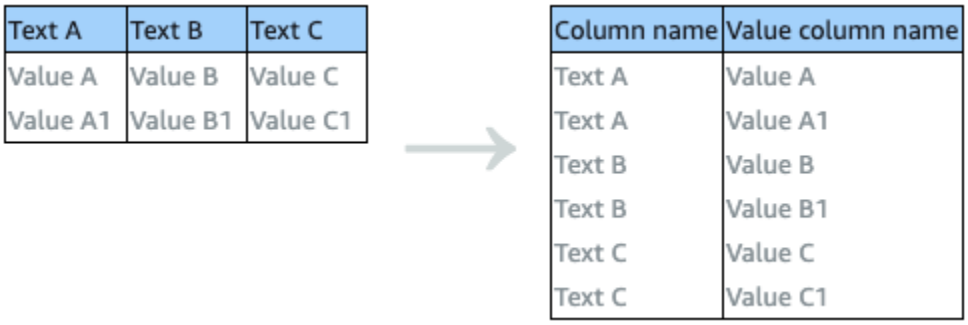
- `datasetsColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste aller Spalten in den Datensätzen darstellt.
- `secondaryDatasetNames`— Eine JSON-encoded Zeichenfolge, die eine Liste von einem oder mehreren sekundären Datensätzen darstellt.
- `secondaryInputs`— Eine JSON-encoded Zeichenfolge, die eine Liste von Amazon S3 S3-Buckets und Objektschlüsselnamen darstellt, die angeben, DataBrew wo sich die sekundären Datensätze befinden.
- `targetColumnNames`— Eine JSON-encoded Zeichenfolge, die eine Liste von Spaltennamen für die Ergebnisse darstellt.

Example Beispiel

```
{
    "Action": {
        "Operation": "UNION",
        "Parameters": {
            "datasetsColumns": "[[\"assembly_session\",\"state_code\",
\"state_name\",\"year\",\"all_votes\",\"yes_votes\",\"no_votes\",\"abstain
\",\"idealpoint_estimate\",\"affinityscore_usa\",\"affinityscore_russia\",
\"affinityscore_china\",\"affinityscore_india\",\"affinityscore_brazil\",
\"affinityscore_israel\"],[\"assembly_session\",\"state_code\",\"state_name
\",null,null,null,null,null,null,null,null,null,null,null,null]]",
            "secondaryDatasetNames": "[\"votes\"]",
            "secondaryInputs": "[{\"S3InputDefinition\":{\"Bucket\":\"databrew-public-
datasets-us-east-1\",\"Key\":\"votes.csv\"}}]",
            "targetColumnNames": "[\"assembly_session\",\"state_code\",\"state_name\",
\"year\",\"all_votes\",\"yes_votes\",\"no_votes\",\"abstain\",\"idealpoint_estimate
\",\"affinityscore_usa\",\"affinityscore_russia\",\"affinityscore_china\",
\"affinityscore_india\",\"affinityscore_brazil\",\"affinityscore_israel\"]"
        }
    }
}
```

UNPIVOT

Konvertiert alle Spaltenwerte in einer ausgewählten Zeile in einzelne Zeilen mit Werten.



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`— Eine JSON-encoded Zeichenfolge, die eine Liste mit einer oder mehreren Spalten darstellt, deren Pivotierung aufgehoben werden soll.
- `unpivotColumn`— Die Wertespalte für den Vorgang zum Entpivotieren.
- `valueColumn`— Die Spalte, die Werte ohne Pivotierung enthält.

Example Beispiel

```
{
    "Action": {
        "Operation": "UNPIVOT",
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
    }
}
```

Rezeptschritte für Datenwissenschaft

Verwenden Sie diese Rezeptschritte, um Daten aus verschiedenen Perspektiven tabellarisch darzustellen und zusammenzufassen oder um erweiterte Transformationen durchzuführen.

Themen

- BINARISIERUNG
- BUCKETISIERUNG
- CATEGORICAL_MAPPING
- ONE_HOT_ENCODING
- SCALE
- SCHIEFHEIT
- TOKENISIERUNG

BINARISIERUNG

Nimmt alle Werte in einer ausgewählten numerischen Quellspalte, vergleicht sie mit einem Schwellenwert und gibt für jede Zeile eine neue Spalte mit einer 1 oder 0 aus.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

  `threshold`— Zahl, die den Schwellenwert für die Zuweisung des Werts 0 oder 1 angibt.

  `flip`— Option zum Umkehren der binären Zuweisung, sodass niedrigeren Werten 1 und höheren Werten 0 zugewiesen wird. Wenn der Umkehrparameter den Wert true hat, ergeben Werte, die kleiner oder gleich dem Schwellenwert sind, den Wert 1 und Werte, die über dem Schwellenwert liegen, den Wert 0.

Example Beispiel

```
{
    "Action": {
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
```

```
}
```

BUCKETISIERUNG

Bei der Bucketisierung (in der Konsole als Einteilung bezeichnet) werden die Elemente in einer Spalte mit numerischen Werten in Abschnitte gruppiert, die durch numerische Bereiche definiert sind, und es wird eine neue Spalte ausgegeben, in der der Abschnitt für jede Zeile angezeigt wird. Die Bucketisierung kann mithilfe von Teilungen oder Prozentwerten erfolgen. Das erste Beispiel unten verwendet Splits und das zweite Beispiel verwendet einen Prozentsatz.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

  `bucketNames`— Liste der Bucket-Namen.

  `splits`— Liste der Bucket-Levels. Buckets sind aufeinander folgend, und eine Obergrenze für einen Bucket ist eine Untergrenze für den nächsten Bucket.

  `percentage`— Jeder Bereich wird als Prozentwert beschrieben.

Example Beispiel für die Verwendung von Splits

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
        }
    }
}
```

Example Beispiel mit einem Prozentsatz

```
{
```

```
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

CATEGORICAL_MAPPING

Ordnet einen oder mehrere kategoriale Werte numerischen oder anderen Werten zu

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `categoryMap`— Eine JSON-encoded Zeichenfolge, die eine Zuordnung von Werten zu Kategorien darstellt.

  `deleteOtherRows`— Fall`true`, werden alle Zeilen, die nicht zugeordnet sind, aus dem Datensatz entfernt.

  `other`— Falls angegeben, werden alle nicht zugewiesenen Werte durch diesen Wert ersetzt.

  `keepOthers`— Falls wahr, bleiben alle nicht zugewiesenen Werte gleich.

  `mapType`— Der Datentyp der zugewiesenen Spalte.

  `targetColumn`— Der Name einer Spalte, die die Ergebnisse enthalten soll.

Example Beispiel

```
{
    "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

Erzeugt n numerische Spalten, wobei n die Anzahl der Einzelwerte in einer ausgewählten kategorialen Variablen ist.

Stellen Sie sich zum Beispiel eine Spalte mit dem Namen `shirt_size` vor. Hemden sind in den Größen S, M, L oder XL erhältlich. Die Spaltendaten könnten wie folgt aussehen.

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
M
```

In diesem Szenario gibt es vier unterschiedliche Werte für`shirt_size`. `ONE_HOT_ENCODING`Generiert daher vier neue Spalten. Jede neue Spalte ist benannt`shirt_size_`*x*, wobei sie *x* für einen bestimmten `shirt_size` Wert steht.

Die Ergebnisse von `shirt_size` und die vier generierten Spalten sehen so aus.

```
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
```

Die Spalte, für die Sie angeben, `ONE_HOT_ENCODING` kann maximal zehn (10) unterschiedliche Werte haben.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte. Die Spalte kann maximal 10 unterschiedliche Werte haben.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SCALE

Skaliert oder normalisiert den Datenbereich in einer numerischen Spalte.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `strategy`— Die Operation, die auf die Spaltenwerte angewendet werden soll:
  - `MIN_MAX`— Skaliert die Werte neu in einen Bereich von [0,1]
  - `SCALE_BETWEEN`— Skaliert die Werte in einen Bereich von 2 angegebenen Werten neu.

- `MEAN_NORMALIZATION`— Skaliert die Daten neu, sodass sie einen Mittelwert (μ) von 0 und eine Standardabweichung (σ) von 1 innerhalb eines Bereichs von [-1, 1] haben
- `Z_SCORE`— Datenwerte werden linear skaliert, sodass sie einen Mittelwert (μ) von 0 und eine Standardabweichung (μ) von 1 haben. Am besten für den Umgang mit Ausreißern geeignet.

- `targetColumn`— Der Name einer Spalte, die die Ergebnisse enthalten soll.

Example Beispiel

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

SCHIEFHEIT

Wendet Transformationen auf Ihre Datenwerte an, um die Form der Verteilung und ihre Neigung zu ändern.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

  `skewFunction`
  - `ROOT`— extrahiert die Wertewurzel. Die Wurzel kann im `value` Parameter angegeben werden.

    `LOG`— Basiswert protokollieren. Die Protokollbasis kann im `value` Parameter angegeben werden.

    `SQUARE`— quadratische Funktion

  `value`— Argument der SkewFunction.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

TOKENISIERUNG

Teilt Text in kleinere Einheiten oder Tokens auf, z. B. einzelne Wörter oder Begriffe.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `delimiter`— Ein benutzerdefiniertes Trennzeichen, das zwischen Wörtern mit einem Token angezeigt wird. (Das Standardverhalten besteht darin, jedes Token durch ein Leerzeichen zu trennen.)
- `expandContractions`— Wenn`ENABLED`, erweitert zusammengefasste Wörter. Zum Beispiel: Aus „nicht“ wird „nicht“.
- `stemmingMode`— Teilt Text in kleinere Einheiten oder Zeichen auf, z. B. einzelne Wörter oder Begriffe in Kleinbuchstaben. Zwei Stemming-Modi sind verfügbar: |. `PORTER` `LANCASTER`
- `stopWordRemovalMode`— Entfernt gebräuchliche Wörter wie a, an, the und mehr.
- `customStopWords`— Für`StopWordRemovalMode`, ermöglicht es Ihnen, eine benutzerdefinierte Liste von Stoppwörtern anzugeben.
- `targetColumn`— Der Name einer Spalte, die die Ergebnisse enthalten soll.

Example Beispiel

```
{
```

```
    "Action": {
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

Mathematische Funktionen

Im Folgenden finden Sie Referenzthemen für mathematische Funktionen, die mit Rezeptaktionen arbeiten.

Themen

- ABSOLUTE
- ADD
- CEILING
- DEGREES
- TEILEN
- EXPONENT
- FLOOR
- IST_GERADE
- IS_ODD
- LN
- LOG
- MOD
- MULTIPLIZIEREN
- NEGIEREN

- PI
- POWER
- RADIANS
- RANDOM
- RANDOM_BETWEEN
- ROUND
- SIGN
- SQUARE_ROOT
- SUBTRAHIEREN

ABSOLUTE

Gibt den absoluten Wert der eingegebenen Zahl in einer neuen Spalte zurück. Der absolute Wert gibt an, wie weit die Zahl von Null entfernt ist, unabhängig davon, ob sie positiv oder negativ ist

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
        }
    }
}
```

ADD

Summiert die Werte der Eingabespalten in einer neuen Spalte mit (`sourceColumn1+sourceColumn2`) oder (`sourceColumn1+value1`).

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `value1`— Ein numerischer Wert.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

CEILING

Gibt die kleinste Ganzzahl zurück, die größer oder gleich den eingegebenen Dezimalzahlen in einer neuen Spalte ist.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value1`— Ein numerischer Wert.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

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

```
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

DEGREES

Konvertiert Radiant für einen Winkel in Grad und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

TEILEN

Dividiert eine eingegebene Zahl durch eine andere und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `value1`— Ein numerischer Wert.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value2`— Ein numerischer Wert.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
        }
    }
}
```

EXPONENT

Gibt die auf n-ten Grad erhöhte Eulersche Zahl in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_EXPONENT"
        }
    }
}
```

FLOOR

Gibt die größte ganze Zahl zurück, die größer oder gleich der eingegebenen Zahl in einer neuen Spalte ist.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `value`— Ein numerischer Wert.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

IST_GERADE

Gibt einen booleschen Wert in einer neuen Spalte zurück, der angibt, ob die Quellspalte oder der Quellwert gerade ist. Wenn die Quellspalte oder der Wert eine Dezimalzahl ist, ist das Ergebnis falsch.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `trueString` – Eine Zeichenfolge, die angibt, ob der Wert gerade ist.
- `falseString`— Eine Zeichenfolge, die angibt, ob der Wert ungerade ist.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
```

```
        "Parameters": {
            "falseString": "Value is odd",
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

IS_ODD

Gibt einen booleschen Wert in einer neuen Spalte zurück, der angibt, ob die Quellspalte oder der Quellwert ungerade ist. Wenn die Quellspalte oder der Wert eine Dezimalzahl ist, ist das Ergebnis falsch.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `trueString`— Eine Zeichenfolge, die angibt, ob der Wert ungerade ist.
- `falseString`— Eine Zeichenfolge, die angibt, ob der Wert nicht ungerade ist.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
}
```

LN

Gibt den natürlichen Logarithmus (Eulersche Zahl) eines Werts in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
        }
    }
}
```

LOG

Gibt den Logarithmus eines Werts in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `base`— Die Basis des Logarithmus. Der Standardwert ist 10.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "LOG",
        "Parameters": {
            "base": "10",
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
```

```
    }
}
```

MOD

Gibt den Prozentsatz zurück, in dem eine Zahl zu einer anderen Zahl in einer neuen Spalte gehört.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

MULTIPLIZIEREN

Multipliziert zwei Zahlen und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `value1`— Ein numerischer Wert.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value2`— Ein numerischer Wert.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

NEGIEREN

Negiert einen Wert und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

Gibt den Wert von Pi (3,141592653589793) in einer neuen Spalte zurück.

Parameters

- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

POWER

Gibt den Wert einer Zahl potenziert mit dem Exponenten in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value`— Eine Zahl, deren Wert erhöht werden soll.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.
- `exponent`— Die Potenz, auf die der Wert erhöht wird.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

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

```
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
    }
}
```

RADIANS

Konvertiert Grad in Radiant (dividiert durch 180/pi) und gibt den Wert in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

RANDOM

Gibt eine Zufallszahl zwischen 0 und 1 in einer neuen Spalte zurück.

Parameters

- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "RANDOM",
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

RANDOM_BETWEEN

Gibt in einer neuen Spalte eine Zufallszahl zwischen einer angegebenen Untergrenze (einschließlich) und einer angegebenen Obergrenze (einschließlich) zurück.

Parameters

- `lowerBound`— Die Untergrenze des Zufallszahlenbereichs.
- `upperBound`— Die obere Grenze des Zufallszahlenbereichs.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

ROUND

Rundet einen numerischen Wert auf die nächste Ganzzahl in einer neuen Spalte ab. Es wird aufgerundet, wenn der Bruch 0,5 oder mehr beträgt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

SIGN

Gibt eine neue Spalte mit -1 zurück, wenn der Wert kleiner als 0 ist, 0, wenn der Wert 0 ist, und +1, wenn der Wert größer als 0 ist.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

SQUARE_ROOT

Gibt die Quadratwurzel eines Werts in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

SUBTRAHIEREN

Subtrahiert eine Zahl von einer anderen und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `value1`— Ein numerischer Wert.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value2`— Ein numerischer Wert.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "targetColumn": "weight_minus_10_kg",
```

```
            "value2": "10"
        }
    }
}
```

Aggregationsfunktionen

Im Folgenden finden Sie Referenzthemen für Aggregatfunktionen, die mit Rezeptaktionen arbeiten.

Themen

- ANY
- AVERAGE
- COUNT
- COUNT_DISTINCT
- KTH_LARGEST
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN
- MODE
- STANDARD_DEVIATION
- SUM
- VARIANCE

ANY

Gibt alle Werte aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Leere Werte und Nullwerte werden ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

AVERAGE

Berechnet den Durchschnitt der Werte in den Quellspalten und gibt das Ergebnis in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

COUNT

Gibt die Anzahl der Werte aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Leere Werte und Nullwerte werden ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

COUNT_DISTINCT

Gibt die Gesamtzahl der unterschiedlichen Werte aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Leere Werte und Nullwerte werden ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "COUNT_DISTINCT",
        "Parameters": {
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
```

```
    }
}
```

KTH_LARGEST

Gibt die k-größte Zahl aus den ausgewählten Quellspalten in einer neuen Spalte zurück.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.
- `value`— Eine Zahl, die k darstellt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

Gibt die k-größte eindeutige Zahl aus den ausgewählten Quellspalten in einer neuen Spalte zurück.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

  `value`— Eine Zahl, die k darstellt.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

Gibt den maximalen numerischen Wert aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
}
```

MEDIAN

Gibt den Median, die mittlere Zahl einer sortierten Zahlengruppe, aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

Gibt den Mindestwert aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird ignoriert.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "MIN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

Gibt den Modus, also die Zahl, die am häufigsten vorkommt, aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird ignoriert. Bei mehreren Modi wird der Modus mit der Modalfunktion berechnet.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

STANDARD_DEVIATION

Gibt die Standardabweichung von den ausgewählten Quellspalten in einer neuen Spalte zurück.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

SUM

Gibt die Summe der Werte aus den ausgewählten Quellspalten in einer neuen Spalte zurück. Jede Zahl, die keine Zahl ist, wird als 0 behandelt.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
```

```
    }
}
```

VARIANCE

Gibt die Varianz der ausgewählten Quellspalten in einer neuen Spalte zurück. Varianz ist definiert als. `Var(X) = [Sum ((X – mean(X))^2)]/Count(X)`

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "VARIANCE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

Textfunktionen

Im Folgenden finden Sie Referenzthemen für Textfunktionen, die mit Rezeptaktionen funktionieren.

Themen

- CHAR
- ENDS_WITH
- EXAKT
- FINDEN
- LEFT
- LEN

- LOWER
- MERGE_COLUMNS_AND_VALUES
- RICHTIG
- REMOVE_SYMBOLS
- REMOVE_WHITESPACE
- REPEAT_STRING
- RIGHT
- RIGHT_FIND
- STARTS_WITH
- STRING_GREATER_THAN
- STRING_GREATER_THAN_EQUAL
- STRING_LESS_THAN
- STRING_LESS_THAN_EQUAL
- SUBSTRING
- TRIM
- UNICODE
- UPPER

CHAR

Gibt in einer neuen Spalte das Unicode-Zeichen für jede Ganzzahl in der Quellspalte oder für einen benutzerdefinierten Ganzzahlwert zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value`— Eine Ganzzahl, die einen Unicode-Wert darstellt.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
}
```

ENDS_WITH

Gibt `true` in einer neuen Spalte zurück, wenn eine angegebene Anzahl von Zeichen ganz rechts oder eine benutzerdefinierte Zeichenfolge einem Muster entspricht.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `pattern`— Ein regulärer Ausdruck, der mit dem Ende der Zeichenfolge übereinstimmen muss.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

EXAKT

Erstellt eine neue Spalte, die mit einer der folgenden Angaben gefüllt ist:

- `True`wenn eine Zeichenfolge in einer Spalte (oder einem Wert) exakt mit einer anderen Zeichenfolge in einer anderen Spalte (oder einem anderen Wert) übereinstimmt.
- `False`wenn es keine Übereinstimmung gibt.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Beide von`sourceColumn`*N*.
> - Einer von `sourceColumn`*N* und einer von`value`*N*.
> - Beide von`value`*N*.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "EXACT",
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

FINDEN

Bei der Suche von links nach rechts werden Zeichenketten gefunden, die einer bestimmten Zeichenfolge aus der Quellspalte oder einem benutzerdefinierten Wert entsprechen, und das Ergebnis wird in einer neuen Spalte zurückgegeben.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Ein regulärer Ausdruck, nach dem gesucht werden soll.
- `position`— Die Position des Zeichens, mit der das Zeichen beginnen soll, vom linken Ende der Zeichenfolge aus.
- `ignoreCase`— Wenn`true`, ignoriere die Unterschiede zwischen Groß- und Kleinschreibung zwischen Buchstaben. Um eine strikte Übereinstimmung zu erzwingen, verwenden `false` Sie stattdessen.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
```

```
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

LEFT

Entnimmt bei gegebener Anzahl von Zeichen die am weitesten links stehende Anzahl von Zeichen in der Zeichenfolge aus der Quellspalte oder der benutzerdefinierten Zeichenfolge und gibt die angegebene Anzahl von Zeichen ganz links in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `position`— Die Position des Zeichens, mit der das Zeichen beginnen soll, vom linken Ende der Zeichenfolge aus.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "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

Gibt in einer neuen Spalte die Länge von Zeichenketten aus der Quellspalte oder von benutzerdefinierten Zeichenketten zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
```

```
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
        }
    }
}
```

LOWER

Konvertiert alle alphabetischen Zeichen aus den Zeichenfolgen in der Quellspalte oder benutzerdefinierten Zeichenketten in Kleinbuchstaben und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
```

```
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
    }
}
```

MERGE_COLUMNS_AND_VALUES

Verkettet die Zeichenketten in den Quellspalten und gibt das Ergebnis in einer neuen Spalte zurück. Sie können ein Trennzeichen zwischen den zusammengeführten Werten einfügen.

Parameters

- `sourceColumns`— Die Namen von zwei oder mehr vorhandenen Spalten im JSON-encoded Format.
- `delimiter` – Optional. Ein oder mehrere Zeichen, die zwischen jeweils zwei Quellspaltenwerten platziert werden sollen.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
```

```
    }
}
```

RICHTIG

Konvertiert alle alphabetischen Zeichen aus den Zeichenfolgen in der Quellspalte oder den benutzerdefinierten Werten in die richtige Groß- und Kleinschreibung und gibt das Ergebnis in einer neuen Spalte zurück.

In der richtigen Schreibweise, auch Großschreibung genannt, wird der erste Buchstabe jedes Worts groß geschrieben und der Rest des Worts in Kleinbuchstaben umgewandelt. Ein Beispiel ist: Der schnelle braune Fuchs sprang über den Zaun

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_proper"
        }
    }
}
```

```
{
```

```
    "RecipeAction": {
        "Operation": "PROPER",
        "Parameters": {
            "value": "MR. H. SMITH, ESQ.",
            "targetColumn": "formal_name_proper"
        }
    }
}
```

REMOVE_SYMBOLS

Entfernt Zeichen, die keine Buchstaben, Zahlen, lateinischen Akzentzeichen oder Leerzeichen sind, aus den Zeichenfolgen in der Quellspalte oder benutzerdefinierten Zeichenfolgen und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "sourceColumn": "info_url",
            "targetColumn": "info_url_remove_symbols"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_SYMBOLS",
        "Parameters": {
            "value": "$&#$&HEY!#@@",
            "targetColumn": "without_symbols"
        }
    }
}
```

REMOVE_WHITESPACE

Entfernt Leerraum aus den Zeichenfolgen in der Quellspalte oder den benutzerdefinierten Zeichenketten und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "sourceColumn": "job_desc",
            "targetColumn": "job_desc_remove_whitespace"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "REMOVE_WHITESPACE",
        "Parameters": {
            "value": "This string has spaces in it",
            "targetColumn": "string_without_spaces"
        }
    }
}
```

REPEAT_STRING

Wiederholt die Zeichenfolgen in der Quellspalte oder dem benutzerdefinierten Eingabewert eine angegebene Anzahl von Malen und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `count`— Gibt an, wie oft die Zeichenfolge wiederholt werden soll.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "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

Entnimmt bei einer bestimmten Anzahl von Zeichen die am weitesten rechts stehende Anzahl von Zeichen in den Zeichenketten aus der Quellspalte oder benutzerdefinierten Zeichenketten und gibt die angegebene Anzahl von Zeichen ganz rechts in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `position`— Die Position des Zeichens, mit der das Zeichen beginnen soll, von der rechten Seite der Zeichenfolge aus.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "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

Bei der Suche von rechts nach links werden Zeichenketten gefunden, die einer bestimmten Zeichenfolge aus der Quellspalte oder einem benutzerdefinierten Wert entsprechen, und das Ergebnis wird in einer neuen Spalte zurückgegeben.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `pattern`— Ein regulärer Ausdruck, nach dem gesucht werden soll.
- `position`— Die Position des Zeichens, mit der das Zeichen beginnen soll, vom rechten Ende der Zeichenfolge aus.
- `ignoreCase`— Wenn`true`, ignoriere die Unterschiede zwischen Groß- und Kleinschreibung zwischen Buchstaben. Um eine strikte Übereinstimmung zu erzwingen, verwenden `false` Sie stattdessen.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

Gibt `true` in einer neuen Spalte zurück, wenn eine angegebene Anzahl von Zeichen ganz links oder eine benutzerdefinierte Zeichenfolge einem Muster entspricht.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `pattern`— Ein regulärer Ausdruck, der mit dem Anfang der Zeichenfolge übereinstimmen muss.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "STARTS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
```

```
        }
    }
}
```

STRING_GREATER_THAN

Erstellt eine neue Spalte, die mit einem der folgenden Werte gefüllt ist:

- `True`wenn eine Zeichenfolge in einer Spalte (oder einem Wert) größer ist als eine andere Zeichenfolge in einer anderen Spalte (oder einem anderen Wert).
- `False`wenn es keine Übereinstimmung gibt.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Beide von`sourceColumn`*N*.
> - Einer von `sourceColumn`*N* und einer von`value`*N*.
> - Beide von`value`*N*.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
```

```
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
        }
    }
}
```

STRING_GREATER_THAN_EQUAL

Erstellt eine neue Spalte, die mit einem der folgenden Werte gefüllt ist:

- `True`wenn eine Zeichenfolge in einer Spalte (oder einem Wert) größer oder gleich einer anderen Zeichenfolge in einer anderen Spalte (oder einem anderen Wert) ist.
- `False`wenn es keine Übereinstimmung gibt.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Beide von`sourceColumn`*N*.
> - Einer von `sourceColumn`*N* und einer von`value`*N*.
> - Beide von`value`*N*.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN_EQUAL",
```

```
        "Parameters": {
            "sourceColumn1": "nationality",
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

STRING_LESS_THAN

Erstellt eine neue Spalte, die mit einem der folgenden Werte gefüllt ist:

- `True`wenn eine Zeichenfolge in einer Spalte (oder einem Wert) kleiner als eine andere Zeichenfolge in einer anderen Spalte (oder einem anderen Wert) ist.
- `False`wenn es keine Übereinstimmung gibt.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Beide von`sourceColumn`*N*.
> - Einer von `sourceColumn`*N* und einer von`value`*N*.
> - Beide von`value`*N*.

Example Beispiel

```
{
```

```
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

STRING_LESS_THAN_EQUAL

Erstellt eine neue Spalte, die mit einem der folgenden Werte gefüllt ist:

- `True`wenn eine Zeichenfolge in einer Spalte (oder einem Wert) kleiner oder gleich einer anderen Zeichenfolge in einer anderen Spalte (oder einem anderen Wert) ist.
- `False`wenn es keine Übereinstimmung gibt.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Beide von`sourceColumn`*N*.
> - Einer von `sourceColumn`*N* und einer von`value`*N*.
> - Beide von`value`*N*.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

Gibt in einer neuen Spalte einige oder alle der angegebenen Zeichenketten in der Quellspalte zurück, basierend auf den benutzerdefinierten Start- und Endindexwerten.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `startPosition`— Die Position des Zeichens, mit der das Zeichen beginnen soll, vom linken Ende der Zeichenfolge aus.
- `endPosition`— Die Zeichenposition, mit der das Zeichen enden soll, vom linken Ende der Zeichenfolge aus.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
```

```
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
}
```

TRIM

Entfernt führende und abschließende Leerzeichen aus den Zeichenfolgen in der Quellspalte oder den benutzerdefinierten Zeichenketten und gibt das Ergebnis in einer neuen Spalte zurück. Leerzeichen zwischen Wörtern werden nicht entfernt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

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

```
        "Operation": "TRIM",
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
    }
}
```

UNICODE

Gibt in einer neuen Spalte den Unicode-Indexwert für das erste Zeichen der Zeichenketten in der Quellspalte oder für benutzerdefinierte Zeichenketten zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
```

```
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
        }
    }
}
```

UPPER

Konvertiert alle alphabetischen Zeichen aus den Zeichenfolgen in der Quellspalte oder benutzerdefinierten Zeichenketten in Großbuchstaben und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
        }
    }
}
```

Datums- und Zeitfunktionen

Im Folgenden finden Sie Referenzthemen für Datums- und Uhrzeitfunktionen, die mit Rezeptaktionen funktionieren.

Themen

- CONVERT_TIMEZONE
- DATE
- DATE_ADD
- DATE_DIFF
- DATUMSFORMAT
- DATE_TIME
- TAG
- STUNDE
- MILLISEKUNDE
- MINUTE
- MONAT
- MONATSNAME
- NOW
- QUARTAL
- SECOND
- TIME
- HEUTE

- UNIX_TIME
- UNIX_TIME_FORMAT
- WOCHENTAG
- WOCHE_NUMMER
- JAHR

CONVERT_TIMEZONE

Konvertiert einen Zeitwert aus der Quellspalte in eine neue Spalte, die auf einer angegebenen Zeitzone basiert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte. Die Quellspalte kann vom Typ `stringdate`, oder `timestamp` sein.
- `fromTimeZone`— Quellwert Zeitzone. Wenn nichts angegeben ist, ist die Standardzeitzone UTC.
- `toTimeZone`— Zeitzone, in die konvertiert werden soll. Wenn nichts angegeben ist, ist die Standardzeitzone UTC.
- `targetColumn`— Ein Name für die neu erstellte Spalte.
- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum. Wenn das Format nicht angegeben ist, wird das Standardformat verwendet:`yyyy-mm-dd HH:MM:SS`.

Example Beispiel

```
{
    "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

Erstellt eine neue Spalte, die den Datumswert aus den Quellspalten oder aus den angegebenen Werten enthält.

Parameters

- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn diese Zeichenfolge nicht angegeben ist, ist das Standardformat`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Eine JSON-encoded Zeichenfolge, die die Komponenten von Datum und Uhrzeit darstellt:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Jede Komponente muss eine der folgenden Angaben enthalten:
  - `sourceColumn` – Der Name einer vorhandenen Spalte.
  - `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "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

Fügt dem Datum aus einer Quellspalte oder einem Quellwert ein Jahr, einen Monat oder einen Tag hinzu und erstellt eine neue Spalte mit den Ergebnissen.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `units`— Eine Maßeinheit für die Anpassung des Datums. Gültige Werte sind `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, und`MINUTES`.
- `dateAddValue`— Die Zahl der`units`, die dem Datum hinzugefügt werden sollen.
- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn nichts angegeben ist, ist das Standardformat `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
```

```
}
```

DATE_DIFF

Erstellt eine neue Spalte, die den Unterschied zwischen zwei Daten enthält.

Parameters

- `sourceColumn1` – Der Name einer vorhandenen Spalte.
- `sourceColumn2` – Der Name einer vorhandenen Spalte.
- `value1` – Eine auszuwertende Zeichenfolge.
- `value2` – Eine auszuwertende Zeichenfolge.
- `units`— Eine Maßeinheit zur Beschreibung der Differenz zwischen den Daten. Gültige Werte sind `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, und`MINUTES`.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können nur eine der folgenden Kombinationen angeben:
>
> - Sowohl von als `sourceColumn1` auch`sourceColumn2`.
> - Einer von `sourceColumn1` oder `sourceColumn2` und einer von `value1` oder`value2`.
> - Sowohl von als `value1` auch`value2`.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
```

```
        }
    }
}
```

DATUMSFORMAT

Erstellt eine neue Spalte mit einem Datum in einem bestimmten Format aus einer Zeichenfolge, die ein Datum darstellt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value`— Eine auszuwertende Zeichenfolge.
- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn nichts angegeben ist, ist das Standardformat `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiele

```
{
    "RecipeAction": {
        "Operation": "DATE_FORMAT",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "dateTimeFormat": "month*dd*yyyy",
            "targetColumn": "DATE Column 1_DATEFORMAT"
        }
    }
}
```

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

```
        "Operation": "DATE_FORMAT",
        "Parameters": {
            "value": "22:10:47",
            "dateTimeFormat": "HH:MM:SS",
            "targetColumn": "formatted_date_value"
        }
    }
}
```

DATE_TIME

Erstellt eine neue Spalte, die den Datums- und Uhrzeitwert aus den Quellspalten oder aus den angegebenen Werten enthält.

Parameters

- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn diese Zeichenfolge nicht angegeben ist, ist das Standardformat`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Eine JSON-encoded Zeichenfolge, die die Komponenten von Datum und Uhrzeit darstellt:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Jede Komponente muss eine der folgenden Angaben enthalten:
  - `sourceColumn` – Der Name einer vorhandenen Spalte.
  - `value` – Eine auszuwertende Zeichenfolge.

Example Beispiel

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

TAG

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte, die den Tag des Monats enthält.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

STUNDE

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte mit dem Stundenwert.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
}
```

MILLISEKUNDE

Erstellt eine neue Spalte, die den Millisekundenwert aus einer Quellspalte oder einem Eingabewert enthält.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte. Die Quellspalte kann vom Typ `string`, `date` oder sein. `timestamp`
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn`— Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
}
```

MINUTE

Erstellt eine neue Spalte, die den Minutenwert enthält, aus einer Zeichenfolge, die ein Datum darstellt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
```

```
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
}
```

MONAT

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte, die die Zahl des Monats enthält.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

MONATSNAME

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte mit dem Namen des Monats.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

NOW

Erstellt eine neue Spalte, die das aktuelle Datum und die aktuelle Uhrzeit im Format enthältyyyy-mm-dd HH:MM:SS.

Parameters

- `timeZone`— Der Name einer Zeitzone. Wenn keine Zeitzone angegeben ist, ist die Standardeinstellung Universal Coordinated Time (UTC).
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

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

```
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
}
```

QUARTAL

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte mit dem datumsbasierten Quartal.

> **Note**
>
> Quartale werden in der neuen Spalte als 1, 2, 3 oder 4 gekennzeichnet.
>
> - 1 steht für Januar, Februar und März.
> - 2 steht für April, Mai und Juni.
> - 3 steht für Juli, August und September.
> - 4 steht für Oktober, November und Dezember.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte. Die Quellspalte kann vom Typ `stringdate`, oder sein`timestamp`.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn`— Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

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

```
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
}
```

SECOND

Erstellt eine neue Spalte, die den zweiten Wert enthält, aus einer Zeichenfolge, die ein Datum darstellt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

Erstellt eine neue Spalte, die den Zeitwert aus den angegebenen Quellspalten oder -werten enthält.

Parameters

- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn diese Zeichenfolge nicht angegeben ist, ist das Standardformat`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— Eine JSON-encoded Zeichenfolge, die die Komponenten von Datum und Uhrzeit darstellt:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Jede Komponente muss eine der folgenden Angaben enthalten:
  - `sourceColumn` – Der Name einer vorhandenen Spalte.
  - `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

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

HEUTE

Erstellt eine neue Spalte, die das aktuelle Datum im Format enthältyyyy-mm-dd.

Parameters

- timeZone— Der Name einer Zeitzone. Wenn keine Zeitzone angegeben ist, ist die Standardeinstellung Universal Coordinated Time (UTC).
- targetColumn – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
        }
    }
}
```

UNIX_TIME

Erstellt eine neue Spalte mit einer Zahl, die die Epochenzeit (Unix-Zeit) — die Anzahl der Sekunden seit dem 1. Januar 1970 — darstellt, basierend auf einer Quellspalte oder einem Eingabewert. Wenn eine Zeitzone abgeleitet werden kann, befindet sich die Ausgabe in dieser Zeitzone. Andernfalls erfolgt die Ausgabe in UTC (Universal Coordinated Time).

Parameters

- sourceColumn – Der Name einer vorhandenen Spalte.
- value – Eine auszuwertende Zeichenfolge.
- targetColumn – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
        }
    }
}
```

UNIX_TIME_FORMAT

Konvertiert die Unix-Zeit für eine Quellspalte oder einen Eingabewert in ein bestimmtes numerisches Datumsformat und gibt das Ergebnis in einer neuen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value`— Eine Ganzzahl, die einen Zeitstempel der Unix-Epoche darstellt.
- `dateTimeFormat` – Optional. Eine Formatzeichenfolge für das Datum, wie es in der neuen Spalte erscheinen soll. Wenn nichts angegeben ist, ist das Standardformat `yyyy-mm-dd HH:MM:SS`.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

WOCHENTAG

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte, die den Wochentag enthält.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

WOCHE_NUMMER

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte, die die Zahl der Woche (von 1 bis 52) enthält.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

JAHR

Erstellt aus einer Zeichenfolge, die ein Datum darstellt, eine neue Spalte mit dem Jahr.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

> **Note**
>
> Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
}
```

Fensterfunktionen

Im Folgenden finden Sie Referenzthemen für Fensterfunktionen, die mit Rezeptaktionen funktionieren.

Themen

- FILL
- NEXT
- ZURÜCK
- ROLLING_AVERAGE
- ROLLING_COUNT_A
- ROLLING_KTH_LARGEST
- ROLLING_KTH_LARGEST_UNIQUE
- ROLLING_MAX
- ROLLING_MIN
- ROLLING_MODE
- ROLLING_STANDARD_DEVIATION
- ROLLING_SUM

- ROLLING_VARIANCE
- ROW_NUMBER
- SESSION

FILL

Gibt eine neue Spalte zurück, die auf einer angegebenen Quellspalte basiert. `FILL`Wählt für alle fehlenden Werte oder Nullwerte in der Quellspalte den neuesten, nicht leeren Wert aus einem Fenster mit Zeilen vor und nach dem betreffenden Quellwert aus. Der gewählte Wert wird dann in die neue Spalte eingefügt.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
}
```

NEXT

Gibt eine neue Spalte zurück, wobei jeder Wert für einen Wert steht, der sich n Zeilen später in der Quellspalte befindet.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- numRows— Ein Wert, der n Zeilen weiter vorne in der Quellspalte darstellt. Wenn beispielsweise 3 numRows ist, wird der drittnächste `sourceColumn` Wert als neuer `targetColumn` Wert NEXT verwendet.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
}
```

ZURÜCK

Gibt eine neue Spalte zurück, wobei jeder Wert einen Wert darstellt, der sich n Zeilen weiter vorne in der Quellspalte befindet.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- numRows— Ein Wert, der n Zeilen weiter vorne in der Quellspalte darstellt. Wenn beispielsweise 3 numRows ist, wird der drittvorherige `sourceColumn` Wert als neuer `targetColumn` Wert PREV verwendet.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

ROLLING_AVERAGE

Gibt in einer neuen Spalte den gleitenden Durchschnitt der Werte von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
```

```
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
    }
}
```

ROLLING_COUNT_A

Gibt in einer neuen Spalte die fortlaufende Anzahl von Werten, die ungleich Null sind, von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

Gibt in einer neuen Spalte den viertgrößten Wert aus einer bestimmten Anzahl von Zeilen vor und einer angegebenen Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `value`— Der Wert für k.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
  "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

Gibt in einer neuen Spalte den k−größten rollierenden eindeutigen Wert von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.

- `value`— Der Wert für k.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
  "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

Gibt in einer neuen Spalte das rollierende Maximum der Werte von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

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

```
        "Operation": "ROLLING_MAX",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

ROLLING_MIN

Gibt in einer neuen Spalte das rollende Minimum der Werte von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

ROLLING_MODE

Gibt in einer neuen Spalte den Rollmodus (gängigster Wert) von einer bestimmten Anzahl von Zeilen vor zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- modeType — Die modale Funktion, die auf das Fenster angewendet werden soll. Gültige Werte sind `NONE`, `MINIMUM`, `MAXIMUM` und `AVERAGE`.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

ROLLING_STANDARD_DEVIATION

Gibt in einer neuen Spalte die gleitende Standardabweichung von Werten von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

ROLLING_SUM

Gibt in einer neuen Spalte die rollierende Summe der Werte von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.

  `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.

- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

ROLLING_VARIANCE

Gibt in einer neuen Spalte die rollierende Varianz der Werte von einer bestimmten Anzahl von Zeilen vor bis zu einer bestimmten Anzahl von Zeilen nach der aktuellen Zeile in der angegebenen Spalte zurück.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `numRowsBefore`— Eine Anzahl von Zeilen vor der aktuellen Quellzeile, die den Anfang des Fensters darstellt.
- `numRowsAfter`— Eine Anzahl von Zeilen nach der aktuellen Quellzeile, die das Ende des Fensters darstellt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
```

```
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

Gibt in einer neuen Spalte eine Sitzungs-ID zurück, die auf einem Fenster basiert, das aus Spaltennamen aus den Anweisungen „group by“ und „order by“ erstellt wurde.

Parameters

- `groupByColumns`— Eine JSON-encoded Zeichenfolge, die die Spalten „Gruppieren nach“ beschreibt.
- `orderByColumns`— Eine JSON-encoded Zeichenfolge, die die „Sortierung nach“ -Spalten beschreibt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESSION

Gibt in einer neuen Spalte eine Sitzungs-ID zurück, die auf einem Fenster basiert, das aus Spaltennamen aus den Anweisungen „group by“ und „order by“ erstellt wurde.

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `units`— Eine Maßeinheit zur Beschreibung der Sitzungsdauer. Gültige Werte sind `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, und`MINUTES`.
- `value`— Die Zahl von`units`, um den Zeitraum zu definieren.
- `groupByColumns`— Eine JSON-encoded Zeichenfolge, die die Spalten „Gruppieren nach“ beschreibt.
- `orderByColumns`— Eine JSON-encoded Zeichenfolge, die die „Sortierung nach“ -Spalten beschreibt.
- `targetColumn` – Ein Name für die neu erstellte Spalte.

Example Beispiel

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
}
```

Web-Funktionen

Im Folgenden finden Sie Referenzthemen für Webfunktionen, die mit Rezeptaktionen funktionieren.

Themen

- IP_TO_INT
- INT_TO_IP
- URL_PARAMS

IP_TO_INT

Konvertiert den IPv4-Wert (Internet Protocol Version 4) der Quellspalte oder einen anderen Wert in den entsprechenden Ganzzahlwert in der Zielspalte und gibt das Ergebnis in einer neuen Spalte zurück. Diese Funktion funktioniert nur für IPv4.

Stellen Sie sich zum Beispiel die folgende IP-Adresse vor.

```
192.168.1.1
```

Wenn Sie diesen Wert als Eingabe für verwenden`IP_TO_INT`, lautet der Ausgabewert wie folgt.

```
3232235777
```

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_TO_IP

Konvertiert den Integer-Wert der Quellspalte oder einen anderen Wert in den entsprechenden IPv4-Wert in der Zielspalte und gibt das Ergebnis in einer neuen Spalte zurück. Diese Funktion funktioniert nur für IPv4.

Betrachten Sie zum Beispiel die folgende Ganzzahl.

```
167772410
```

Wenn Sie diesen Wert als Eingabe für verwenden`INT_TO_IP`, lautet der Ausgabewert wie folgt.

```
10.0.0.250
```

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
[  {
    "RecipeAction": {
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
  }
]
```

URL_PARAMS

Extrahiert Abfrageparameter aus einer URL-Zeichenfolge, formatiert sie als JSON-Objekt und gibt das Ergebnis in einer neuen Spalte zurück.

Stellen Sie sich zum Beispiel die folgende URL vor.

```
https://example.com/?firstParam=answer&secondParam=42
```

Wenn Sie diesen Wert als Eingabe für verwendenURL_PARAMS, lautet der Ausgabewert wie folgt.

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parameters

- `sourceColumn` – Der Name einer vorhandenen Spalte.
- `value` – Eine auszuwertende Zeichenfolge.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Sie können `sourceColumn` oder `value` angeben, aber nicht beides.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
    }
}
```

Andere Funktionen

Im Folgenden finden Sie Referenzthemen für andere Funktionen, die mit Rezeptaktionen funktionieren.

Themen

- COALESCE
- GET_ACTION_RESULT
- GET_STEP_DATAFRAME

COALESCE

Gibt in einer neuen Spalte den ersten Wert im Spaltenarray zurück, der ungleich Null ist. Die Reihenfolge der in der Funktion aufgelisteten Spalten bestimmt die Reihenfolge, in der sie durchsucht werden.

Parameters

- `sourceColumns`— Eine JSON-encoded Zeichenfolge, die eine Liste vorhandener Spalten darstellt.
- `targetColumn` – Der Name der neuen Spalte, die erstellt werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

Ruft das Ergebnis einer zuvor übermittelten Aktion ab. Nur zur Verwendung im interaktiven Erlebnis.

Parameters

- `actionId`— Das ActionId wurde in der ursprünglichen SendProjectSessionAction Antwort zurückgegeben.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "GET_ACTION_RESULT",
        "Parameters": {
            "actionId": "7",
        }
    }
}
```

GET_STEP_DATAFRAME

Ruft den Datenrahmen aus einem Schritt im Projektrezept ab. Nur zur Verwendung im interaktiven Erlebnis. Wird zusammen mit dem ViewFrame Parameter verwendet, um einen großen Datenrahmen zu paginieren.

Parameters

- `stepIndex`— Der Index des Schritts im Projektrezept, für den der Datenrahmen abgerufen werden soll.

Example Beispiel

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

Kontingente für AWS Glue DataBrew

Sie können Ihre DataBrew Service Quotas in der AWS Service Quotas-Konsole einsehen. Sie können auch eine Erhöhung des Kontingents für jedes konfigurierbare Kontingent beantragen.

Dokumenthistorie für AWS Glue DataBrew Entwicklerhandbuch

Aktuelle API-Version: databrew-2017-07-25

In der folgenden Tabelle wird die Dokumentation für diese Version von beschrieben.AWS Glue DataBrew Wenn Sie über die Aktualisierung des AWS Glue DataBrew Entwicklerhandbuchs informiert werden möchten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
`glue:GetCustomEntityType` zu AWS verwalteten Richtlinien hinzugefügt	Diese Berechtigung ist erforderlich, um AWS Glue DataBrew Profiljobs mit PII-identification aktivierter Option auszuführen. Weitere Informationen finden Sie unter AWS Glue DataBrew Aktualisierungen AWS verwalteter Richtlinien.	20. März 2024
Support für mehrere Hash-Algorithmen in der CRYPTOGRAPHIC_HASH-Transformation	Sie können jetzt beim Hashing von Werten in einer Spalte einen Hash-Algorithmus angeben. Weitere Informationen finden Sie unter CRYPTOGRAPHIC_HASH.	11. August 2023
`glue:BatchGetCustomEntityTypes` zu verwalteten Richtlinien hinzugefügt AWS	Diese Berechtigung ist erforderlich, um AWS Glue DataBrew Profiljobs mit PII-identification aktivierter Option auszuführen. Weitere Informationen finden Sie unter AWS Glue DataBrew Aktualisi	9. Mai 2022

	erungen AWS verwalteter Richtlinien.	
Support für das Apache ORC-Dateiformat	DataBrew unterstützt jetzt Apache ORC als Dateiform at für DataBrew Datenquel len und Ausgaben. Weitere Informationen finden Sie unter Unterstützte Dateitypen für Datenquellen.	31. März 2022
Support für kontoüber greifenden AWS Glue Data Catalog Amazon S3 S3-Zugriff	Sie können jetzt von anderen aus auf AWS Glue Data Catalog S3-Tabellen zugreifen ,AWS-Konten wenn in der AWS Glue Konsole eine entsprechende Ressource nrichtlinie erstellt wurde. Nach dem Erstellen einer Richtlini e können die entsprechenden Data Catalog S3-Tabellen bei der Erstellung eines DataBrew Datensatzes als Eingabequellen ausgewählt werden. Weitere Informationen finden Sie unter Unterstützte Verbindungen für Datenquel len und Ausgaben.	11. März 2022

Support für die native Konsolenintegration mit Amazon AppFlow	DataBrew hat jetzt eine native Konsolenintegration mit Amazon AppFlow. Diese Integration bedeutet, dass Sie eine Verbindung zu Daten von Salesforce, Zendesk, Slack und anderen Software-as-a-Service (SaaS) -Anwendungen herstellen können. ServiceNo w Sie können auch eine Verbindung zu Daten von AWS-Services Amazon S3 und Amazon Redshift herstelle n. Weitere Informationen finden Sie unter Unterstützte Verbindungen für Datenquel len und Ausgaben.	18. November 2021
Support von Datenqual itätsregeln	DataBrew unterstützt jetzt die Erstellung von Datenqual itätsregeln, bei denen es sich um anpassbare Validieru ngsprüfungen handelt, die Geschäftsanforderungen für bestimmte Daten definieren. Weitere Informationen finden Sie unter Überprüfen der Datenqualität in AWS Glue DataBrew.	18. November 2021

Support für benutzerdefinierte SQL-Anweisungen	DataBrew unterstützt jetzt benutzerdefinierte SQL-Anweisungen zum Abrufen von Daten aus Amazon Redshift und Snowflake. Diese Unterstützung bedeutet, dass Sie eine speziell entwickelte Abfrage verwenden können, um die aus großen Tabellen zurückgegebenen Daten auszuwählen und einzuschränken. Weitere Informationen finden Sie unter Unterstützte Verbindungen für Datenquellen und Ausgaben.	18. November 2021
Support für PII-Erkennung	DataBrew unterstützt jetzt die Erkennung von personenbezogenen Daten (PII). Dies gibt Ihnen die Möglichkeit, personenbezogene Daten während der Datenaufbereitung zu maskieren. Weitere Informationen finden Sie unter Identifizierung und Umgang mit personenbezogenen Daten (PII).	18. November 2021
Support für weitere AWS Regionen	DataBrew unterstützt jetzt weitere AWS Regionen. Eine Liste der unterstützten Regionen finden Sie unter AWS Glue DataBrew Endpunkte und Kontingente.	5. Oktober 2021

Support für das Schreiben von Daten in Lake Formation -based Amazon S3 S3-Tabellen	DataBrew unterstützt jetzt das Schreiben von Daten in AWS Glue Data Catalog S3-Tabellen auf der Grundlage von AWS Lake Formation . DataBrew unterstützt jetzt auch das Schreiben von Daten in das Tableau Hyper-Format. Weitere Informationen finden Sie unter AWS Glue DataBrew Rezeptjobs erstellen und damit arbeiten.	13. August 2021
Support für das Schreiben von Daten in JDBC-Ziele	DataBrew unterstützt jetzt das direkte Schreiben von Daten in JDBC-supported Datenbanken und Data Warehouses. Dazu gehören Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database und PostgreSQL. Weitere Informationen finden Sie unter Rezeptjobs erstellen und damit arbeiten.AWS Glue DataBrew	23. Juli 2021
Support für die Angabe, welche Datenqualitätsstatistiken für einen Profiljob generiert werden	DataBrew unterstützt jetzt die Angabe, welche Datenqualitätsstatistiken für Datensätze in einem Profiljob automatisch generiert werden. Weitere Informationen finden Sie unter AWS Glue DataBrew Rezeptjobs erstellen und mit ihnen arbeiten.	23. Juli 2021

Support für das Schreiben von Datensätzen in die AWS Glue Data Catalog	DataBrew beinhaltet jetzt Unterstützung für das direkte Schreiben von Datensätzen in die.AWS Glue Data Catalog Sie können wählen, ob Datensätze, die aus Jobs erstellt wurden, die Ihre Datenvorbereitungsrezepte ausführen, in Amazon S3-, Amazon Redshift- und Amazon RDS-Tabellen im Datenkatalog gespeichert werden. Zu den unterstützten RDS-Tabellen gehören die für Amazon Aurora, RDS für Oracle, RDS für Microsoft SQL Server, RDS für MySQL und RDS für PostgreSQL.	30. Juni 2021
Support für die Identifizierung fortgeschrittener Datentypen	DataBrew beinhaltet jetzt Unterstützung für die automatische Identifizierung und Markierung erweiterter Datentypen für Spalten, wodurch es einfacher wird, Spalten zu normalisieren, die bestimmte Datentypen enthalten. Zu diesen Datentypen gehören Sozialversicherungsnummer, E-Mail-Adresse, Telefonnummer, Geschlecht, Kreditkarte, URL, IP-Adresse, Datum und Uhrzeit, Währung, Postleitzahl, Land, Region, Bundesstaat und Stadt.	30. Juni 2021

Support für die Verwendung von Amazon AppFlow zur Übertragung von Daten aus SAAS-Anwendungen	DataBrew unterstützt jetzt die Verwendung von Amazon AppFlow für die Übertragung von Daten aus Software-as-a-Service (SaaS) -Anwendungen (SaaS) von Drittanbietern wie Salesforce, Zendesk, Slack und. ServiceNow Weitere Informationen finden Sie unter Unterstützte Verbindungen für Datenquellen und Ausgaben.	29. April 2021
Support für die Erstellung von DataBrew Datensätzen mit Eingaben aus JDBC-Datenbanken	DataBrew unterstützt jetzt die Erstellung von Datensätzen aus Daten in JDBC-supported Datenbanken und Data Warehouses, darunter Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database und PostgreSQL. Weitere Informationen finden Sie unter Unterstützte Verbindungen für Datenquellen und Ausgaben.	2. April 2021
Support für zusätzliche AWS-Regionen	DataBrew unterstützt jetzt zusätzliche AWS-Regionen. Eine Liste der unterstützten Regionen finden Sie unter AWS Glue DataBrew Endpunkte und Kontingente.	28. Januar 2021

Neue Transformationen für den Umgang mit Duplikaten	Vier neue Transformationen für den Umgang mit Duplikaten wurden der Konsole und der DataBrew API hinzugefügt. Weitere Informationen finden Sie unter DELETE_DUPLICATE_ROWS, FLAG_DUPLICATE_ROWS, FLAG_DUPLICATES_IN_COLUMNund REMOVE_DUPLICATES in den Rezeptschritten zur Datenqualität.	28. Januar 2021
Zusätzliche CSV-Trennzeichen	DataBrew unterstützt jetzt neben Kommas weitere Trennzeichen in Dateien mit kommagetrennten Werten (CSV), die zur Erstellung von Datensätzen verwendet werden. DataBrew Weitere Informationen finden Sie unter Datensätze erstellen und verwenden.AWS Glue DataBrew	28. Januar 2021
DataBrew Erweiterung für JupyterLab	Jetzt können Sie es AWS Glue DataBrew als Erweiterung in verwenden JupyterLab. Weitere Informationen finden Sie unter DataBrew Als Erweiterung verwenden in JupyterLab.	20. November 2020
Neues Tool zur Datenaufbereitung:AWS Glue DataBrew	Dies ist die erste Version des AWS Glue DataBrew-Entwicklerhandbuchs.	11. November 2020

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.