aws

Unable to locate subtitle

AWS Glue DataBrew Panduan Developer

AWS Glue DataBrew Panduan Developer: ***Unable to locate subtitle***

Table of Contents

Apa itu DataBrew? 1
Konsep dan istilah inti 2
Proyek 3
Set data 3
Resep 3
Lowongan 4
Silsilah data 4
Profil data 4
Integrasi produk dan layanan 4
Penyiapan 7
Menyiapkan yang baru AWS akun 7
Menyiapkan AWS CLI 9
Menyiapkan izin Peran IAM 10
Menyiapkan kebijakan IAM untuk DataBrew 11
Menambahkan pengguna dan grup dengan DataBrew izin 25
Menambahkan peran IAM dengan izin DataBrew 26
Penyiapan AWS IAM Identity Center(Pusat Identitas IAM) 26
Langkah-langkah login untuk pengguna IAM Identity Center-enabled 28
Menggunakan DataBrew di JupyterLab 28
Prasyarat 29
Mengkonfigurasi JupyterLab untuk menggunakan ekstensi 31
Mengaktifkan DataBrew ekstensi untuk JupyterLab 33
Memulai 35
Prasyarat 35
Langkah 1: Buat proyek 35
Langkah 2: Ringkas data 36
Langkah 3: Tambahkan lebih banyak transformasi 37
Langkah 4: Tinjau DataBrew sumber daya Anda 38
Langkah 5: Buat profil data 39
Langkah 6: Ubah kumpulan data 40
Langkah 7: (Opsional) Bersihkan 42
Set data 43
Jenis file yang didukung untuk sumber data 43
Koneksi yang didukung untuk sumber data dan output 45

Menggunakan dataset ... 49
Menghapus dataset ... 53
Menghubungkan ke data Anda ... 53
Menggunakan driver JDBC untuk menghubungkan data ... 54
Driver JDBC yang didukung ... 56
Menghubungkan ke data dalam file teks dengan DataBrew ... 57
Menghubungkan data dalam beberapa file di Amazon S3 ... 59
Skema saat menggunakan banyak file sebagai kumpulan data ... 59
Menggunakan jalur berparameter untuk Amazon S3 ... 60
Jenis Data ... 71
Tipe data tingkat lanjut ... 72
Tipe data tingkat lanjut ... 72
Memvalidasi kualitas data ... 74
Memvalidasi aturan kualitas data ... 75
Bertindak atas hasil validasi ... 75
Membuat ruleset dengan aturan kualitas data ... 76
Membuat pekerjaan profil ... 78
Memeriksa hasil validasi untuk dan memperbarui aturan kualitas data ... 79
Cek yang tersedia ... 80
Proyek ... 98
Membuat proyek ... 99
Ikhtisar sesi DataBrew proyek ... 100
Tampilan kisi ... 101
Tampilan skema ... 103
Tampilan profil ... 104
Menghapus proyek ... 107
Resep ... 108
Menerbitkan versi resep baru ... 109
Mendefinisikan struktur resep ... 109
Ketentuan penggunaan ... 113
Lowongan ... 116
Lowongan kerja resep ... 116
Contoh partisi kolom ... 121
Mengotomatiskan pekerjaan berjalan dengan jadwal ... 121
Bekerja dengan ekspresi cron untuk pekerjaan resep ... 122
Menghapus pekerjaan dan jadwal pekerjaan ... 126

Lowongan kerja profil 126
Membangun konfigurasi pekerjaan profil secara terprogram 128
Keamanan 143
Perlindungan data 144
Enkripsi saat diam 145
Enkripsi saat bergerak 148
Manajemen kunci 148
Mengidentifikasi dan menangani PII 149
DataBrew ketergantungan pada layanan lain AWS 150
Manajemen identitas dan akses 150
Mengautentikasi dengan identitas 151
Mengelola akses menggunakan kebijakan 152
AWS Glue DataBrew and AWS Lake Formation 154
Bagaimana AWS Glue DataBrew bekerja dengan IAM 154
Identity-based contoh kebijakan 158
AWS Kebijakan Terkelola untuk DataBrew 162
Pemecahan masalah 167
Pencatatan log dan pemantauan 169
Validasi kepatuhan 169
Ketahanan 170
Keamanan infrastruktur 170
Penggunaan AWS Glue DataBrew dengan VPC Anda 171
Penggunaan AWS Glue DataBrew dengan titik akhir VPC 172
Analisis konfigurasi dan kerentanan di AWS Glue DataBrew 172
Pemantauan DataBrew 173
Pemantauan CloudWatch dengan 174
Mengotomatisasi dengan Acara CloudWatch 174
Pemantauan dengan CloudWatch Log 177
Logging panggilan API dengan CloudTrail 177
DataBrew Informasi di CloudTrail 177
Memahami Entri File DataBrew Log 178
Penggunaan AWS Pemberitahuan Pengguna dengan AWS Glue Databrew 179
Langkah resep dan referensi fungsi 181
Langkah-langkah resep kolom dasar 183
CHANGE_DATA_TYPE 184
DELETE 185

MENGGANDAKAN ... 185
JSON_TO_STRUCTS ... 186
BERGERAK_SETELAH ... 187
BERGERAK_SEBELUM ... 187
MOVE_TO_END ... 188
MOVE_TO_INDEX ... 188
MOVE_TO_START ... 189
GANTI NAMA ... 189
SORT ... 190
TO_BOOLEAN_COLUMN ... 191
KE_DOUBLE_COLUMN ... 192
TO_NUMBER_COLUMN ... 192
TO_STRING_COLUMN ... 193
Langkah-langkah resep pembersihan data ... 194
CAPITAL_CASE ... 195
FORMAT_DATE ... 195
HURUF KECIL ... 196
UPPER_CASE ... 196
SENTENCE_CASE ... 197
ADD_DOUBLE_QUOTES ... 197
ADD_PREFIX ... 198
ADD_SINGLE_QUOTES ... 198
ADD_SUFFIX ... 199
EXTRACT_BETWEEN_DELIMITERS ... 199
EXTRACT_BETWEEN_POSITIONS ... 200
EXTRACT_PATTERN ... 201
EXTRACT_VALUE ... 201
REMOVE_COMBINED ... 203
REPLACE_BETWEEN_DELIMITERS ... 206
REPLACE_BETWEEN_POSITIONS ... 207
REPLACE_TEXT ... 208
Langkah-langkah resep kualitas data ... 209
ADVANCED_DATATYPE_FILTER ... 210
ADVANCED_DATATYPE_FLAG ... 211
DELETE_DUPLICATE_ROWS ... 212
EXTRACT_ADVANCED_DATATYPE_DETAILS ... 213

FILL_WITH_AVERAGE 214
FILL_WITH_CUSTOM 214
FILL_WITH_EMPTY 215
FILL_WITH_LAST_VALID 215
FILL_WITH_MEDIAN 216
FILL_WITH_MODE 216
FILL_WITH_MOST_FREQUENT 217
FILL_WITH_NULL 218
FILL_WITH_SUM 218
FLAG_DUPLICATE_ROWS 219
FLAG_DUPLICATES_IN_COLUMN 219
GET_ADVANCED_DATATYPE 220
HAPUS_DUPLIKAT 220
REMOVE_INVALID 221
REMOVE_MISSING 222
REPLACE_WITH_AVERAGE 222
REPLACE_WITH_CUSTOM 223
REPLACE_WITH_EMPTY 224
REPLACE_WITH_LAST_VALID 224
REPLACE_WITH_MEDIAN 225
REPLACE_WITH_MODE 225
REPLACE_WITH_MOST_FREQUENT 226
REPLACE_WITH_NULL 227
REPLACE_WITH_ROLLING_AVERAGE 227
REPLACE_WITH_ROLLING_SUM 228
REPLACE_WITH_SUM 229
Langkah resep PII 229
CRYPTOGRAPHIC_HASH 230
MENDEKRIPSI 232
DETERMINISTIC_DECRYPT 233
DETERMINISTIC_ENCRYPT 234
MENGENKRIPSI 235
MASK_CUSTOM 237
MASK_DATE 237
TOPENG_PEMBATAS 238
MASK_RANGE 239

REPLACE_WITH_RANDOM_BETWEEN ... 240
REPLACE_WITH_RANDOM_DATE_BETWEEN ... 241
SHUFFLE_ROWS ... 241
Deteksi outlier dan langkah-langkah penanganan resep ... 242
FLAG_OUTLIER ... 242
REMOVE_OUTLIERS ... 244
REPLACE_OUTLIERS ... 246
RESCALE_OUTLIERS_WITH_Z_SCORE ... 249
RESCALE_OUTLIERS_WITH_SKEW ... 251
Langkah-langkah resep struktur kolom ... 253
BOOLEAN_OPERASI ... 254
CASE_OPERATION ... 270
FLAG_COLUMN_FROM_NULL ... 283
FLAG_COLUMN_FROM_PATTERN ... 284
MERGE ... 284
SPLIT_COLUMN_BETWEEN_DELIMITER ... 285
SPLIT_COLUMN_BETWEEN_POSITIONS ... 286
SPLIT_COLUMN_FROM_END ... 286
SPLIT_COLUMN_FROM_START ... 287
SPLIT_COLUMN_MULTIPLE_DELIMITER ... 287
SPLIT_COLUMN_SINGLE_DELIMITER ... 288
SPLIT_COLUMN_WITH_INTERVAL ... 289
Langkah resep pemformatan kolom ... 289
NOMOR_FORMAT ... 290
FORMAT_PHONE_NUMBER ... 291
Langkah-langkah resep struktur data ... 293
NEST_TO_ARRAY ... 293
NEST_TO_MAP ... 294
NEST_TO_STRUCT ... 295
UNNEST_ARRAY ... 295
UNNEST_MAP ... 296
UNNEST_STRUCT ... 297
UNNEST_STRUCT_N ... 297
GROUP_BY ... 298
BERGABUNG ... 299
POROS ... 300

SKALA ... 301
MENTRANSPOS ... 302
UNION ... 303
UNPIVOT ... 304
Langkah-langkah resep ilmu data ... 305
BINARISASI ... 305
BUCKETIZATION ... 306
CATEGORICAL_MAPPING ... 307
ONE_HOT_PENGKODEAN ... 308
SKALA ... 301
KEMIRINGAN ... 310
TOKENISASI ... 311
Fungsi matematika ... 312
MUTLAK ... 313
MENAMBAHKAN ... 314
LANGIT-LANGIT ... 314
GELAR ... 315
MEMBAGI ... 315
EKSPONEN ... 316
FLOOR ... 317
ADALAH_GENAP ... 317
ADALAH_ANEH ... 318
LN ... 319
LOG ... 319
MOD ... 320
KALIKAN ... 320
MENIADAKAN ... 321
PI ... 321
DAYA ... 322
RADIAN ... 323
ACAK ... 323
RANDOM_BETWEEN ... 324
BULAT ... 324
TANDA ... 325
SQUARE_ROOT ... 325
KURANGI ... 326

Fungsi agregat 327
APA PUN 327
RATA-RATA 328
COUNT 328
COUNT_DISTINCT 329
KTH_LARGEST 330
KTH_LARGEST_UNIQUE 330
MAX 331
MEDIAN 331
MIN 332
MODE 332
STANDARD_DEVIASI 333
JUMLAH 334
PERBEDAAN 334
Fungsi teks 335
CHAR 336
ENDS_WITH 337
PASTI 337
TEMUKAN 338
KIRI 339
LEN 340
LEBIH RENDAH 341
MERGE_COLUMNS_AND_VALUES 342
TEPAT 343
REMOVE_SYMBOLS 344
REMOVE_WHITESPACE 345
REPEAT_STRING 346
KANAN 347
RIGHT_FIND 348
STARTS_WITH 349
STRING_GREATER_THAN 350
STRING_GREATER_THAN_EQUAL 351
STRING_LESS_THAN 352
STRING_LESS_THAN_EQUAL 353
SUBSTRING 354
MEMANGKAS 355

UNICODE 356
ATAS 357
Fungsi tanggal dan waktu 358
CONVERT_TIMEZONE 359
DATE 359
DATE_ADD 360
DATE_DIFF 361
DATE_FORMAT 362
DATE_TIME 363
DAY 364
HOUR 365
MILIDETIK 366
MINUTE 367
MONTH 367
BULAN_NAMA 368
SEKARANG 369
KUARTAL 369
DETIK 370
TIME 371
HARI INI 372
UNIX_WAKTU 373
FORMAT UNIX_TIME_ 373
WEEK_DAY 374
WEEK_NUMBER 375
YEAR 376
Fungsi jendela 376
FILL 377
SELANJUTNYA 378
SEBELUMNYA 378
ROLLING_AVERAGE 379
ROLLING_COUNT_A 380
ROLLING_KTH_LARGEST 380
ROLLING_KTH_LARGEST_UNIQUE 381
ROLLING_MAX 382
BERGULING_MIN 382
ROLLING_MODE 383

ROLLING_STANDARD_DEVIATION 384
ROLLING_SUM 385
ROLLING_VARIANCE 385
ROW_NUMBER 386
SESI 387
Fungsi web 387
IP_TO_INT 388
INT_TO_IP 389
URL_PARAMS 389
Fungsi lainnya 390
BERSATU 391
GET_ACTION_RESULT 391
GET_STEP_DATAFRAME 392
Kuota dan batasan 393
Riwayat dokumen 394
AWS Glosarium 402
..... cdiii

Apa itu AWS Glue DataBrew?

AWS Glue DataBrew adalah alat persiapan data visual yang memungkinkan pengguna untuk membersihkan dan menormalkan data tanpa menulis kode apa pun. Menggunakan DataBrew membantu mengurangi waktu yang dibutuhkan untuk menyiapkan data untuk analitik dan pembelajaran mesin (ML) hingga 80 persen, dibandingkan dengan persiapan data yang dikembangkan secara khusus. Anda dapat memilih dari lebih dari 250 transformasi siap pakai untuk mengotomatiskan tugas persiapan data, seperti memfilter anomali, mengonversi data ke format standar, dan mengoreksi nilai yang tidak valid.

Menggunakan DataBrew, analis bisnis, ilmuwan data, dan insinyur data dapat lebih mudah berkolaborasi untuk mendapatkan wawasan dari data mentah. Karena DataBrew tanpa server, apa pun tingkat teknis Anda, Anda dapat menjelajahi dan mengubah terabyte data mentah tanpa perlu membuat cluster atau mengelola infrastruktur apa pun.

Dengan DataBrew antarmuka intuitif, Anda dapat secara interaktif menemukan, memvisualisasikan, membersihkan, dan mengubah data mentah. DataBrew membuat saran cerdas untuk membantu Anda mengidentifikasi masalah kualitas data yang sulit ditemukan dan memakan waktu untuk diperbaiki. Dengan DataBrew menyiapkan data Anda, Anda dapat menggunakan waktu Anda untuk bertindak berdasarkan hasil dan mengulangi lebih cepat. Anda dapat menyimpan transformasi sebagai langkah dalam resep, yang dapat Anda perbarui atau gunakan kembali nanti dengan kumpulan data lain, dan terapkan secara berkelanjutan.

Gambar berikut menunjukkan cara DataBrew kerja pada tingkat tinggi.

Untuk menggunakan DataBrew, Anda membuat proyek dan terhubung ke data Anda. Di ruang kerja proyek, Anda melihat data Anda ditampilkan dalam antarmuka visual seperti kisi. Di sini, Anda dapat menjelajahi data dan melihat distribusi nilai dan bagan untuk memahami profilnya.

Untuk menyiapkan data, Anda dapat memilih dari lebih dari 250 transformasi point-and-click. Ini termasuk menghapus null, mengganti nilai yang hilang, memperbaiki inkonsistensi skema, membuat kolom berdasarkan fungsi, dan banyak lagi. Anda juga dapat menggunakan transformasi untuk menerapkan teknik pemrosesan bahasa alami (NLP) untuk membagi kalimat menjadi frasa. Pratinjau langsung menunjukkan sebagian data Anda sebelum dan sesudah transformasi, sehingga Anda dapat memodifikasi resep Anda sebelum menerapkannya ke seluruh kumpulan data.

DataBrew Setelah menjalankan resep Anda di dataset Anda, output disimpan di Amazon Simple Storage Service (Amazon S3). Setelah kumpulan data Anda dibersihkan dan disiapkan ada di Amazon S3, penyimpanan data atau sistem manajemen data Anda yang lain dapat menelannya.

Konsep dan istilah inti dalam AWS Glue DataBrew

Berikut ini, Anda dapat menemukan ikhtisar konsep inti dan terminologi di AWS Glue DataBrew. Setelah Anda membaca bagian ini, lihatMemulai dengan AWS Glue DataBrew, yang memandu Anda melalui proses pembuatan proyek dan menghubungkan kumpulan data dan menjalankan pekerjaan.

Topik

- Proyek
- Set data
- Resep
- Pekerjaan
- Silsilah data
- Profil data

Proyek

Ruang kerja persiapan data interaktif DataBrew disebut proyek. Menggunakan proyek data, Anda mengelola kumpulan item terkait: data, transformasi, dan proses terjadwal. Sebagai bagian dari pembuatan proyek, Anda memilih atau membuat kumpulan data untuk dikerjakan. Selanjutnya, Anda membuat resep, yang merupakan serangkaian instruksi atau langkah yang DataBrew ingin Anda lakukan. Tindakan ini mengubah data mentah Anda menjadi formulir yang siap dikonsumsi oleh pipeline data Anda.

Set data

Dataset berarti sekumpulan data—baris atau catatan yang dibagi menjadi kolom atau bidang. Saat Anda membuat DataBrew proyek, Anda terhubung ke atau mengunggah data yang ingin Anda ubah atau persiapkan. DataBrew dapat bekerja dengan data dari sumber apa pun, diimpor dari file yang diformat, dan terhubung langsung ke daftar penyimpanan data yang terus bertambah.

Untuk DataBrew, dataset adalah koneksi read-only ke data Anda. DataBrew mengumpulkan satu set metadata deskriptif untuk merujuk ke data. Tidak ada data aktual yang dapat diubah atau disimpan oleh DataBrew. Untuk mempermudah, kami menggunakan dataset untuk merujuk pada dataset aktual dan penggunaan DataBrew metadata.

Resep

Dalam DataBrew, resep adalah serangkaian instruksi atau langkah untuk data yang DataBrew ingin Anda tindaklanjuti. Resep dapat berisi banyak langkah, dan setiap langkah dapat berisi banyak tindakan. Anda menggunakan alat transformasi pada bilah alat untuk mengatur semua perubahan yang ingin Anda lakukan pada data Anda. Kemudian, ketika Anda siap untuk melihat produk jadi dari resep Anda, Anda menetapkan pekerjaan ini DataBrew dan menjadwalkannya. DataBrew menyimpan instruksi tentang transformasi data, tetapi tidak menyimpan data Anda yang

sebenarnya. Anda dapat mengunduh dan menggunakan kembali resep di proyek lain. Anda juga dapat mempublikasikan beberapa versi resep.

Pekerjaan

DataBrew mengambil tugas mengubah data Anda dengan menjalankan instruksi yang Anda atur ketika Anda membuat resep. Proses menjalankan instruksi ini disebut pekerjaan. Pekerjaan dapat menempatkan resep data Anda ke dalam tindakan sesuai dengan jadwal yang telah ditetapkan. Tetapi Anda tidak terbatas pada jadwal. Anda juga dapat menjalankan pekerjaan sesuai permintaan. Jika Anda ingin membuat profil beberapa data, Anda tidak memerlukan resep. Dalam hal ini, Anda hanya dapat mengatur pekerjaan profil untuk membuat profil data.

Silsilah data

DataBrew melacak data Anda dalam antarmuka visual untuk menentukan asalnya, yang disebut garis keturunan data. Tampilan ini menunjukkan kepada Anda bagaimana data mengalir melalui entitas yang berbeda dari tempat asalnya. Anda dapat melihat asalnya, entitas lain yang dipengaruhi olehnya, apa yang terjadi padanya dari waktu ke waktu, dan di mana ia disimpan.

Profil data

Saat Anda memprofilkan data Anda, DataBrew buat laporan yang disebut profil data. Ringkasan ini memberi tahu Anda tentang bentuk data Anda yang ada, termasuk konteks konten, struktur data, dan hubungannya. Anda dapat membuat profil data untuk kumpulan data apa pun dengan menjalankan pekerjaan profil data.

Integrasi produk dan layanan

Gunakan bagian ini untuk mengetahui produk dan layanan mana yang terintegrasi DataBrew.

DataBrew bekerja dengan AWS layanan berikut untuk jaringan, manajemen, dan tata kelola:

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

DataBrew bekerja dengan danau AWS data dan penyimpanan data berikut:

- AWS Lake Formation
- Amazon S3

DataBrew mendukung format file dan ekstensi berikut untuk mengunggah data.

Format	Ekstensi file (opsional)	Ekstensi untuk file terkompresi (wajib)

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

Buku kerja Microsoft Excel	`.xlsx`	Tidak ada dukungan kompresi

JSON (dokumen JSON dan baris JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

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

DataBrew menulis file output ke Amazon S3, dan mendukung format dan ekstensi file berikut.

Format	Ekstensi file (tidak terkompresi)	Ekstensi file (terkompresi)
Comma-separated nilai	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated nilai	`.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 Parket	Tidak didukung	`.glue.parquet.snappy`
Apache Avro	`.avro`	`.avro.snappy` , `.avro.gz`, `.avro.lz4` , `.avro.bz2` , `.avro.deflate` , `.avro.br`
Apache ORC	`.orc`	`.orc.snappy` , `.orc.lzo`, `.orc.zlib`
XML	`.xml`	`.xml.snappy` , `.xml.gz`, `.xml.lz4`, `.xml.bz2`, `.xml.deflate` , `.xml.br`
JSON (format JSON Lines saja)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tablo Hyper	Tidak didukung	Tidak berlaku

Menyiapkan AWS Glue DataBrew

Sebelum memulai AWS Glue DataBrew, Anda perlu mengatur beberapa izin, pengguna, dan peran. Mulailah dengan melakukan langkah-langkah berikut:

1. Mendaftar untuk AWS akun sesuai kebutuhan, dan membuat kebijakan AWS Identity and Access Management(IAM) untuk memungkinkan pengguna menjalankan DataBrew:
   - Mendaftar untuk AWS akun baru dan menambahkan pengguna. Untuk informasi selengkapnya, lihat Menyiapkan yang baru AWS akun.
   - Menambahkan kebijakan IAM untuk pengguna konsol. Seorang pengguna dengan izin ini dapat mengakses DataBrew pada file.Konsol Manajemen AWS
   - Menambahkan izin untuk sumber daya data untuk peran IAM. Peran IAM dengan izin ini dapat mengakses data atas nama pengguna.

   Anda harus menjadi administrator IAM untuk membuat pengguna, peran, dan kebijakan.
2. Menambahkan pengguna atau grup untuk DataBrew. Pengguna atau grup dengan izin yang benar terpasang dapat mengakses DataBrew di konsol.
3. Menambahkan peran dengan izin untuk mengakses data untuk DataBrew. Peran dengan izin yang benar dapat mengakses data atas nama pengguna.

Menyiapkan yang baru AWS akun

Jika Anda tidak memiliki AWS akun, daftar AWS akun dan buat pengguna admin IAM.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai

praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Untuk membuat pengguna administrator, pilih salah satu opsi berikut.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
Di Pusat Identitas IAM (Direkomendasikan)	Gunakan kredensi jangka pendek untuk mengakses.AWS Ini sejalan dengan praktik terbaik keamanan. Untuk informasi tentang praktik terbaik, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.	Mengikuti petunjuk di Memulai di Panduan AWS IAM Identity Center Pengguna.	Konfigurasikan akses terprogram dengan Mengonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center dalam AWS Command Line Interface Panduan Pengguna.
Di IAM (Tidak direkomendasikan)	Gunakan kredensi jangka panjang untuk mengakses.AWS	Mengikuti petunjuk di Buat pengguna IAM untuk akses darurat di Panduan Pengguna IAM.	Konfigurasikan akses terprogram dengan Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna IAM:

- Apa itu IAM?
- Menyiapkan dengan IAM
- Membuat pengguna dan grup administrasi (konsol)

Menyiapkan AWS CLI

Jika Anda berencana untuk menggunakan JupyterLab atau DataBrew API, pastikan untuk menginstal AWS Command Line Interface(AWS CLI). Anda tidak membutuhkannya untuk menggunakan DataBrew konsol atau melakukan langkah-langkah dalam latihan Memulai.

Untuk mengatur AWS CLI

1. Unduh dan konfigurasikan AWS CLI dengan menggunakan langkah-langkah yang ditemukan berikut:

   - Menginstal AWS CLI
   - Dasar-dasar Konfigurasi
2. Verifikasi pengaturan dengan memasukkan DataBrew perintah berikut pada prompt perintah.

   ```
   aws databrew help
   ```

   Jika pernyataan ini mengembalikan kesalahan "`aws: error: argument command: Invalid choice`" diikuti oleh daftar panjang layanan, hapus instalan AWS CLI, dan kemudian instal ulang. Tindakan ini tidak menimpa konfigurasi Anda yang ada.

AWS CLI perintah menggunakan AWS Region default dari konfigurasi Anda, kecuali jika Anda mengaturnya dengan parameter atau profil. Anda dapat menambahkan `--region` parameter ke setiap perintah.

Jika mau, Anda dapat menambahkan profil bernama di `~/.aws/config` atau `%UserProfile %/.aws/config` (di Microsoft Windows). Profil bernama juga dapat mempertahankan pengaturan lain, seperti yang ditunjukkan pada contoh berikut.

```
[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
```

Pengaturan AWS Identity and Access Management Izin (IAM)

Sebelum memulai, Anda perlu mengatur beberapa hal di IAM. Anda harus menjadi administrator atau mendapat bantuan dari salah satunya. Namun, jika Anda memiliki akun dengan akses administrator, Anda dapat melakukan tugas-tugas ini sendiri. Anda dapat menemukan instruksi sederhana untuk setiap tugas di bagian ini.

Berikut ini adalah ikhtisar tentang apa yang perlu Anda lakukan:

- Sebagai bagian dari proses ini, Anda menambahkan pengguna. Anda tidak perlu menambahkan pengguna baru, Anda dapat menggunakan yang sudah ada. Anda melampirkan DataBrew izin sehingga pengguna dapat membuka DataBrew konsol.
- Buat peran IAM. Peran memungkinkan tindakan tertentu dan memberikan izin saat digunakan, dalam batas. Misalnya, ini hanya berfungsi untuk pengguna di AWS akun Anda. Anda dapat menambahkan lebih banyak batasan nanti.
- Buat kebijakan atau kebijakan IAM yang Anda butuhkan. Kebijakan adalah daftar hal-hal yang boleh dilakukan pengguna. Untuk membuat kebijakan, Anda membuka halaman konsol lain dan menempelkan teks dari file yang Anda unduh.

> **Note**
>
> Apa yang kami sediakan di sini adalah informasi pengaturan dasar. Kami menyarankan Anda meluangkan waktu untuk menyesuaikan izin Anda sehingga mereka memenuhi kebutuhan keamanan dan kepatuhan Anda. Jika Anda memerlukan bantuan, hubungi administrator atau AWS Support Anda.

Untuk menambahkan izin yang diperlukan

1. Buat kebijakan IAM untuk memungkinkan pengguna menjalankan DataBrew dengan melakukan hal berikut:

   - Tambahkan kebijakan IAM khusus untuk pengguna konsol. Jika Anda tidak memerlukan kebijakan khusus, Anda dapat memilih kebijakan AWS-managed sebagai gantinya.

Tambahkan saja ke pengguna di langkah 2. Pengguna dengan izin ini dapat mengakses konsol DataBrew layanan.

- Tambahkan izin untuk sumber daya data. Peran IAM dengan izin ini dapat mengakses data atas nama pengguna.

Anda harus menjadi administrator untuk membuat pengguna, peran, dan kebijakan.

2. Tambahkan pengguna atau grup untuk DataBrew. Pengguna atau grup dengan izin yang benar terpasang dapat mengakses DataBrew konsol.
3. Tambahkan peran dengan izin untuk mengakses data. DataBrew Peran dengan izin yang benar dapat mengakses data atas nama pengguna.

Menyiapkan kebijakan IAM untuk DataBrew

Anda menggunakan kebijakan IAM untuk mengelola izin. Kebijakan memudahkan untuk menambahkan izin terkait sekaligus, bukan satu per satu.

Kami menyarankan Anda membuat kebijakan menggunakan nama yang sama yang kami berikan. Kami menggunakan nama yang ditampilkan berikut untuk kebijakan ini di seluruh dokumentasi. Menggunakan nama-nama ini juga memudahkan jika Anda perlu menghubungi AWS Support. Namun, Anda dapat memilih untuk mengubah nama kebijakan dan isinya. Untuk informasi selengkapnya tentang kebijakan IAM, lihat Membuat kebijakan yang dikelola pelanggan di Panduan Pengguna IAM.

Setelah membuat kebijakan yang diperlukan untuk digunakan DataBrew, Anda melampirkannya ke pengguna dan peran. Cara melakukan ini dibahas nanti di bagian ini.

Topik

- Menambahkan kebijakan IAM untuk pengguna konsol
- Menambahkan izin untuk sumber daya data untuk peran IAM
- Mengkonfigurasi kebijakan IAM untuk DataBrew

Menambahkan kebijakan IAM untuk pengguna konsol

Menyiapkan izin untuk pengguna Konsol Manajemen AWS adalah opsional, tetapi jika Anda memerlukan akses konsol, ambil langkah ini terlebih dahulu.

Untuk mengatur izin untuk mencapai DataBrew di konsol, pilih salah satu dari berikut ini:

- Gunakan kebijakan yang dikelola oleh AWS:`AwsGlueDataBrewFullAccessPolicy`. Jika Anda memilih opsi ini, lewati ke kebijakan berikutnyaMenambahkan izin untuk sumber daya data untuk peran IAM.
- Buat kebijakan yang dijelaskan di bagian ini,`AwsGlueDataBrewCustomUserPolicy`. Opsi ini memungkinkan Anda untuk menyesuaikan kebijakan dengan persyaratan keamanan khusus tambahan.

Kebijakan berikut memberikan izin yang diperlukan untuk menjalankan konsol. DataBrew Anda memberikan izin tersebut dengan menggunakan IAM.

Untuk menentukan kebijakan AwsGlueDataBrewCustomUserPolicy IAM untuk DataBrew (konsol)

1. Unduh JSON untuk kebijakan `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
3. Di panel navigasi, pilih Kebijakan.
4. Untuk setiap kebijakan, pilih Buat Kebijakan.
5. Di layar Buat Kebijakan, arahkan ke tab JSON.
6. Salin pernyataan kebijakan JSON yang Anda unduh. Tempelkan di atas pernyataan sampel di editor.
7. Verifikasi bahwa kebijakan tersebut disesuaikan dengan akun Anda, persyaratan keamanan, dan AWS sumber daya yang diperlukan. Jika Anda perlu membuat perubahan, Anda dapat membuatnya di editor.
8. Pilih Tinjau kebijakan.

Untuk menentukan kebijakan AwsGlueDataBrewCustomUserPolicy IAM untuk DataBrew (AWS CLI)

1. Unduh JSON untuk kebijakan `AwsGlueDataBrewCustomUserPolicy`IAM.
2. Sesuaikan kebijakan seperti yang dijelaskan pada langkah pertama dari prosedur sebelumnya.
3. Jalankan perintah berikut untuk membuat kebijakan.

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

Menambahkan izin untuk sumber daya data untuk peran IAM

Untuk terhubung ke data,AWS Glue DataBrew perlu memiliki peran IAM yang dapat diteruskan atas nama pengguna. Berikut ini, Anda dapat menemukan cara membuat kebijakan yang nantinya Anda lampirkan ke peran IAM.

`AwsGlueDataBrewDataResourcePolicy`Kebijakan memberikan izin yang diperlukan untuk menyambung ke data menggunakan. DataBrew Untuk setiap operasi yang mengakses data di AWS sumber daya lain, seperti mengakses objek Anda di Amazon S3 DataBrew , memerlukan izin untuk mengakses sumber daya atas nama Anda.

Untuk menentukan kebijakan AwsGlueDataBrewDataResourcePolicy IAM untuk DataBrew (konsol)

1. Unduh JSON untuk `AwsGlueDataBrewDataResourcePolicy`.
2. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
3. Di panel navigasi, pilih Kebijakan.
4. Untuk setiap kebijakan, pilih Buat Kebijakan.
5. Di layar Buat Kebijakan, arahkan ke tab JSON.
6. Salin pernyataan kebijakan JSON yang Anda unduh. Tempelkan di atas pernyataan sampel di editor.
7. Verifikasi bahwa kebijakan tersebut disesuaikan dengan akun Anda, persyaratan keamanan, dan AWS sumber daya yang diperlukan. Jika Anda perlu membuat perubahan, Anda dapat membuatnya di editor.
8. Pilih Tinjau kebijakan.

Untuk menentukan kebijakan AwsGlueDataBrewDataResourcePolicy IAM untuk DataBrew (AWS CLI)

1. Unduh JSON untuk `AwsGlueDataBrewDataResourcePolicy`.
2. Sesuaikan kebijakan seperti yang dijelaskan pada langkah pertama dari prosedur sebelumnya.

3. Jalankan perintah berikut untuk membuat kebijakan.

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

Mengkonfigurasi kebijakan IAM untuk DataBrew

Berikut ini, Anda dapat menemukan detail dan contoh tentang kebijakan IAM yang dapat Anda gunakan. DataBrew Rincian tentang kebijakan dasar disediakan di sini. Plus, ada lebih banyak contoh yang tidak diperlukan untuk digunakan DataBrew. Mereka adalah konfigurasi tambahan yang mungkin Anda gunakan dalam situasi tertentu.

Topik

- AwsGlueDataBrewCustomUserPolicy
- AwsGlueDataBrewDataResourcePolicy
- Kebijakan IAM untuk menggunakan objek Amazon S3 dengan DataBrew
- Kebijakan IAM untuk menggunakan enkripsi dengan DataBrew

AwsGlueDataBrewCustomUserPolicy

`AwsGlueDataBrewCustomUserPolicy`Kebijakan ini memberikan sebagian besar izin yang diperlukan untuk menggunakan konsol. DataBrew Beberapa sumber daya yang ditentukan dalam kebijakan ini mengacu pada layanan yang digunakan oleh DataBrew. Ini termasuk nama untuk AWS Glue Data Catalog, bucket Amazon S3, CloudWatch Log Amazon, dan sumber daya.AWS KMS Hal ini mirip dengan kebijakan AWS-managed bernama`AwsGlueDataBrewFullAccessPolicy`.

Tabel berikut menjelaskan izin yang diberikan oleh kebijakan ini.

Tindakan	Sumber Daya	Deskripsi
`"databrew:*"`	`"*"`	Memberikan izin untuk menjalankan semua operasi DataBrew API.

`"glue:GetDatabases"`
 	
`"glue:GetPartitions"`
	`"*"`	Memungkinkan daftar AWS Glue database dan tabel.

Tindakan	Sumber Daya	Deskripsi
"glue:GetTable" "glue:GetTables" "glue:GetDataCatalogEncryptionSettings"		
"dataexchange:ListDataSets" "dataexchange:ListDataSetRevisions" "dataexchange:ListRevisionAssets" "dataexchange:CreateJob" "dataexchange:StartJob" "dataexchange:GetJob"	"*"	Memungkinkan daftar sumber daya AWS Data Exchange dalam kumpulan data.
"kms:DescribeKey" "kms:ListKeys" "kms:ListAliases"	"*"	Memungkinkan daftar AWS KMS kunci untuk digunakan untuk enkripsi output pekerjaan.
"kms:GenerateDataKey"	"arn:aws:kms:::key/key_ids"	Memungkinkan enkripsi output pekerjaan.

Tindakan	Sumber Daya	Deskripsi
"s3:ListAllMyBuckets" "s3:GetBucketCORS" "s3:GetBucketLocation" "s3:GetEncryptionConfiguration"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Mengizinkan daftar bucket Amazon S3 untuk proyek, kumpulan data, dan pekerjaan. Memungkinkan pengiriman file output ke S3.
"sts:GetCallerIdentity"	"*"	Dapatkan informasi tentang penelepon saat ini.
"cloudtrail:LookupEvents",	"*"	Izinkan AWS CloudTrail acara daftar untuk kumpulan data (garis keturunan data).
"iam:ListRoles" "iam:GetRole"	"*"	Memungkinkan daftar peran IAM untuk digunakan untuk proyek dan pekerjaan.

AwsGlueDataBrewDataResourcePolicy

`AwsGlueDataBrewDataResourcePolicy`Kebijakan memberikan izin yang diperlukan untuk menyambung ke data dan mengonfigurasi. DataBrew

Tabel berikut menjelaskan izin yang diberikan oleh kebijakan ini.

Tindakan	Sumber Daya	Deskripsi
"s3:GetObject"	"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"	Memungkinkan Anda untuk melihat pratinjau file Anda.
"s3:PutObject" "s3:PutBucketCORS"	"arn:aws:s3:::bucket_name/*",	Memungkinkan pengiriman file output ke S3.

Tindakan	Sumber Daya	Deskripsi
	`"arn:aws:s3:::bucket_name"`	
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*"`, `"arn:aws:s3:::bucket_name"`	Memungkinkan menghapus objek yang dibuat oleh DataBrew.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*"`, `"arn:aws:s3:::bucket_name"`	Mengizinkan daftar bucket Amazon S3 dari proyek, kumpulan data, dan pekerjaan.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Memungkinkan dekripsi untuk kumpulan data terenkripsi.
`"kms:GenerateDataKey"`	`"arn:aws:kms:::key/key_ids"`	Memungkinkan enkripsi output pekerjaan.

Tindakan	Sumber Daya	Deskripsi
"ec2:DescribeVpcEndpoints" "ec2:DescribeRouteTables" "ec2:DeleteNetworkInterface" "ec2:DescribeNetworkInterfaces" "ec2:DescribeSecurityGroups" "ec2:DescribeSubnets" "ec2:DescribeVpcAttribute" "ec2:CreateNetworkInterface"	"*"	Memungkinkan penyiapan item jaringan Amazon EC2, seperti virtual private cloud (VPC), saat menjalankan pekerjaan dan proyek.
"ec2:DeleteNetworkInterface"	"*"	Memungkinkan menghapus antarmuka jaringan di VPC.

Tindakan	Sumber Daya	Deskripsi

`"ec2:CreateTags"`
 	
`"ec2:DeleteTags"`
	`"arn:aws:ec2:::network-interface/*", "arn:aws:ec2:::security-group/*"`	Memungkinkan membuat dan menghapus tag.
		
		Anda memerlukan izin ini jika Anda menggunakan Katalog AWS Glue Data dengan VPC diaktifkan. DataBrew meneruskan data AWS Glue untuk menjalankan pekerjaan dan proyek Anda. Izin ini memungkinkan penandaan sumber daya Amazon EC2 yang dibuat untuk titik akhir pengembangan.AWS Glue menandai antarmuka jaringan Amazon EC2, grup keamanan, dan instans dengan. `aws-glue-service-resource`

`"logs:CreateLogGroup"`
 	
`"logs:CreateLogStream"`
 	
`"logs:PutLogEvents"`
	`"arn:aws:logs:::log-group:/aws-glue-databrew/*"`	Memungkinkan menulis log ke Amazon CloudWatch Logs
		
		DataBrew menulis log ke grup log yang namanya dimulai dengan`aws-glue-databrew` .

Tindakan	Sumber Daya	Deskripsi
`"lakeformation:GetDataAccess"`	`"*"`	Memungkinkan akses ke AWS Lake Formation, `"Glue":"GetTable"` disediakan juga diperbolehkan Menggunakan Lake Formation membutuhkan konfigurasi lebih lanjut di konsol Lake Formation.

Kebijakan IAM untuk menggunakan objek Amazon S3 dengan DataBrew

`AwsGlueDataBrewSpecificS3BucketPolicy`Kebijakan ini memberikan izin yang diperlukan untuk mengakses S3 atas nama pengguna non-administratif.

Sesuaikan kebijakan sebagai berikut:

1. Ganti jalur Amazon S3 dalam kebijakan agar jalur tersebut mengarah ke jalur yang ingin Anda gunakan. Dalam teks sampel, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* mewakili objek atau file tertentu. *`BUCKET-NAME-2/`*mewakili semua objek (`*`) yang nama jalurnya dimulai dengan`BUCKET-NAME-2/`. Perbarui ini untuk memberi nama ember yang Anda gunakan.
2. (Opsional) Gunakan wildcard di jalur Amazon S3 untuk membatasi izin lebih lanjut. Untuk informasi lebih lanjut, lihat Elemen kebijakan IAM: Variabel dan tanda dalam Panduan Pengguna IAM.

   Praktik Terbaik Keamanan: Untuk mencegah akses tidak sah ke bucket Amazon S3 dengan nama serupa di akun AWS lain, sertakan kunci kondisi dalam `aws:ResourceAccount` kebijakan Anda. Ini memastikan bahwa hanya DataBrew dapat mengakses bucket dalam AWS akun Anda sendiri, bahkan saat menggunakan ARN sumber daya wildcard. Tambahkan kondisi berikut ke pernyataan kebijakan Anda:

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

Ganti `123456789012` dengan ID AWS akun Anda yang sebenarnya.

Sebagai bagian dari melakukan ini, Anda dapat membatasi izin untuk tindakan `s3:PutObject` dan. `s3:PutBucketCORS` Tindakan ini hanya diperlukan untuk pengguna yang membuat DataBrew proyek, karena pengguna tersebut harus dapat mengirim file output ke S3.

Untuk informasi selengkapnya dan untuk melihat beberapa contoh yang dapat Anda tambahkan ke kebijakan IAM untuk Amazon S3, lihat Contoh Kebijakan Bucket di Panduan Pengembang Amazon S3.

Tabel berikut menjelaskan izin yang diberikan oleh kebijakan ini.

Tindakan	Sumber Daya	Deskripsi
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Memungkinkan Anda untuk melihat pratinjau file Anda.
`"s3:PutObject"` `"s3:PutBucketCORS"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Memungkinkan pengiriman file output ke S3.
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Memungkinkan menghapus objek.

Untuk menentukan kebijakan AwsGlueDataBrewSpecificS3BucketPolicy IAM untuk DataBrew (konsol)

1. Unduh JSON untuk kebijakan `AwsGlueDataBrewSpecificS3BucketPolicy`IAM.
2. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.

3. Di panel navigasi, pilih Kebijakan.
4. Untuk setiap kebijakan, pilih Buat Kebijakan.
5. Di layar Buat Kebijakan, arahkan ke tab JSON.
6. Tempelkan pernyataan JSON kebijakan di atas pernyataan sampel di editor.
7. Verifikasi bahwa kebijakan tersebut disesuaikan dengan akun Anda, persyaratan keamanan, dan AWS sumber daya yang diperlukan. Jika Anda perlu membuat perubahan, Anda dapat membuatnya di editor.
8. Pilih Tinjau kebijakan.

Untuk menentukan kebijakan AwsGlueDataBrewSpecificS3BucketPolicy IAM untuk DataBrew (AWS CLI)

1. Unduh JSON untuk `AwsGlueDataBrewSpecificS3BucketPolicy`.
2. Sesuaikan kebijakan seperti yang dijelaskan pada langkah pertama dari prosedur sebelumnya.
3. Jalankan perintah berikut untuk membuat kebijakan.

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

Kebijakan IAM untuk menggunakan enkripsi dengan DataBrew

`AwsGlueDataBrewS3EncryptedPolicy`Kebijakan ini memberikan izin yang diperlukan untuk mengakses objek S3 yang dienkripsi dengan AWS Key Management Service(AWS KMS) atas nama pengguna non-administratif.

Sesuaikan kebijakan sebagai berikut:

1. Ganti jalur Amazon S3 dalam kebijakan sehingga jalur tersebut mengarah ke jalur yang ingin Anda gunakan. Dalam teks sampel, *`BUCKET-NAME-1/SPECIFIC-OBJECT-NAME`* mewakili objek atau file tertentu. *`BUCKET-NAME-2/`*mewakili semua objek (*) yang nama jalurnya dimulai dengan`BUCKET-NAME-2/`. Perbarui ini untuk memberi nama ember yang Anda gunakan.
2. (Opsional) Gunakan wildcard di jalur Amazon S3 untuk membatasi izin lebih lanjut. Untuk informasi selengkapnya, lihat elemen kebijakan IAM: Variabel dan tag.

Sebagai bagian dari melakukan ini, Anda dapat membatasi izin untuk tindakan `s3:PutObject` dan. `s3:PutBucketCORS` Tindakan ini hanya diperlukan untuk pengguna yang membuat DataBrew proyek, karena pengguna tersebut harus dapat mengirim file output ke S3.

Untuk informasi selengkapnya dan untuk melihat beberapa contoh yang dapat Anda tambahkan ke kebijakan IAM untuk Amazon S3, lihat Contoh Kebijakan Bucket.

3. Temukan ARN sumber daya berikut dalam `ToUseKms` file.

```
  "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. Ubah AWS akun contoh ke nomor AWS akun Anda (tanpa tanda hubung).
5. Ubah daftar sampel untuk mencantumkan peran IAM yang ingin Anda gunakan. Kami menyarankan untuk mencantumkan kebijakan IAM Anda ke set izin sekecil mungkin. Namun, Anda dapat mengizinkan pengguna untuk mengakses semua peran IAM, misalnya jika Anda menggunakan akun pembelajaran pribadi dengan data sampel. Untuk mengizinkan daftar mengakses semua peran IAM, ubah daftar sampel menjadi satu entri:`"arn:aws:iam::111122223333:role/*"`.

Tabel berikut menjelaskan izin yang diberikan oleh kebijakan ini.

Tindakan	Sumber Daya	Deskripsi
`"s3:GetObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Memungkinkan Anda untuk melihat pratinjau file Anda.
`"s3:ListBucket"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Mengizinkan daftar bucket Amazon S3 dari proyek, kumpulan data, dan pekerjaan.
`"s3:PutObject"`	`"arn:aws:s3:::bucket_name/*",`	Memungkinkan pengiriman file output ke S3.

Tindakan	Sumber Daya	Deskripsi
	`"arn:aws:s3:::bucket_name"`	
`"s3:DeleteObject"`	`"arn:aws:s3:::bucket_name/*", "arn:aws:s3:::bucket_name"`	Memungkinkan menghapus objek yang dibuat oleh DataBrew.
`"kms:Decrypt"`	`"arn:aws:kms:::key/key_ids"`	Memungkinkan dekripsi untuk kumpulan data terenkripsi.
`"kms:GenerateDataKey*"`	`"arn:aws:kms:::key/key_ids"`	Memungkinkan enkripsi output pekerjaan.

Untuk menentukan kebijakan AwsGlueDataBrewS3EncryptedPolicy IAM untuk DataBrew (konsol)

1. Unduh JSON untuk kebijakan `AwsGlueDataBrewS3EncryptedPolicy`IAM.
2. Masuk ke Konsol Manajemen AWS dan buka konsol IAM di https://console.aws.amazon.com/iam/.
3. Di panel navigasi, pilih Kebijakan.
4. Untuk setiap kebijakan, pilih Buat Kebijakan.
5. Di layar Buat Kebijakan, arahkan ke tab JSON.
6. Tempelkan pernyataan JSON kebijakan di atas pernyataan sampel di editor.
7. Verifikasi bahwa kebijakan tersebut disesuaikan dengan akun Anda, persyaratan keamanan, dan AWS sumber daya yang diperlukan. Jika Anda perlu membuat perubahan, Anda dapat membuatnya di editor.
8. Pilih Tinjau kebijakan.

Untuk menentukan kebijakan AwsGlueDataBrewS3EncryptedPolicy IAM untuk DataBrew (AWS CLI)

1. Unduh JSON untuk `AwsGlueDataBrewS3EncryptedPolicy`.

2. Sesuaikan kebijakan seperti yang dijelaskan pada langkah pertama dari prosedur sebelumnya.
3. Jalankan perintah berikut untuk membuat kebijakan.

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

Menambahkan pengguna atau grup dengan DataBrew izin

Anda menetapkan kebijakan untuk peran, dan peran untuk pengguna dan grup untuk mengelola izin. Untuk informasi selengkapnya, lihat Identitas IAM (pengguna, grup, dan peran) di Panduan Pengguna IAM.

Sebelum memulai, Anda harus memiliki setidaknya satu pengguna untuk menetapkan izin.

Gunakan prosedur berikut untuk mengatur DataBrew izin bagi pengguna yang perlu bekerja di DataBrew konsol, atau menjalankan DataBrew perintah di CLI.

Untuk mengatur DataBrew izin

1. Buat kunci akses bagi pengguna Anda untuk menggunakan AWS CLI for DataBrew, dan alat pengembangan lainnya.
2. Aktifkan Konsol Manajemen AWS akses untuk memungkinkan pengguna menggunakan AWS konsol.
3. Buat peran untuk DataBrew pengguna atau grup.
4. Pilih kebijakan yang Anda gunakan. Lakukan salah satu tindakan berikut:
   - Jika Anda membuat`AwsGlueDataBrewCustomUserPolicy`, pilih dari daftar.
   - Untuk menggunakan AWS-managed kebijakan, pilih `AwsGlueDataBrewFullAccessPolicy` dari daftar.
5. Tetapkan kebijakan itu untuk peran tersebut.
6. Tetapkan hubungan Kepercayaan untuk peran tersebut sehingga pengguna atau grup dapat mengambil peran yang relevan.
   - Jika Anda tidak menggunakan grup, percayakan pengguna dengan peran tersebut.
   - Jika Anda menggunakan grup, percayakan grup dengan peran tersebut dan tambahkan pengguna ke grup.

Menambahkan peran IAM dengan izin sumber daya data

Anda menggunakan peran IAM untuk mengelola kebijakan yang ditetapkan bersama. Peran IAM dapat digunakan oleh seseorang yang bertindak dalam peran tertentu, seperti DataBrew pengguna atau DataBrew dirinya sendiri. Untuk informasi lebih lanjut, lihat Peran IAM dalam Panduan Pengguna IAM.

Gunakan prosedur berikut untuk membuat peran IAM yang diperlukan untuk DataBrew proyek untuk mengakses data.

Untuk melampirkan kebijakan IAM yang diperlukan ke peran IAM baru DataBrew

1. Dalam panel navigasi, pilih Roles (Peran), Create role (Buat Peran).
2. Untuk Pilih jenis entitas tepercaya, pilih AWS layanan berlabel kartu.
3. Pilih DataBrewdari daftar, lalu pilih Berikutnya: Izin.
4. Masukkan **`AwsGlueDataBrewDataResourcePolicy`** di kotak pencarian (kebijakan IAM yang Anda buat di langkah sebelumnya). Pilih kebijakan dan pilih Berikutnya: Tag.
5. Pilih Berikutnya: Tinjauan.
6. Untuk nama Peran, masukkan**`AwsGlueDataBrewDataAccessRole`**, dan pilih Buat peran.

Penyiapan AWS IAM Identity Center(Pusat Identitas IAM)

Menggunakan AWS IAM Identity Center(IAM Identity Center), pengguna Anda dapat masuk DataBrew dengan URL sederhana, tanpa masuk ke Konsol Manajemen AWS dan tanpa memerlukan akun AWS.

Untuk mengatur Pusat Identitas IAM

1. Buka AWS Organizations konsol, dan buat organisasi jika Anda belum memilikinya. Semua fitur diaktifkan secara default untuk organisasi ini.

   Untuk informasi selengkapnya, lihat AWS IAM Identity Center Prasyarat dan Membuat dan mengelola organisasi.
2. Buka konsol AWS IAM Identity Center
3. Pilih sumber identitas Anda.

Secara default, Anda mendapatkan toko IAM Identity Center untuk manajemen pengguna yang cepat dan mudah. Secara opsional, Anda dapat menghubungkan penyedia identitas eksternal sebagai gantinya, atau menghubungkan AWS Managed Microsoft AD direktori dengan Active Directory lokal. Dalam panduan ini, kami menggunakan toko IAM Identity Center default.

Untuk informasi selengkapnya, lihat Memilih sumber identitas Anda di Panduan AWS IAM Identity Center Pengguna.

4. Buat set izin untuk DataBrew akses:

   a. Di panel navigasi Pusat Identitas IAM, pilih AWS akun, lalu pilih Set izin.

   b. Pada halaman Buat set izin, pilih Buat set izin khusus.

   c. Untuk status Relay, masukkan`https://console.aws.amazon.com/databrew/home?region=us-east-1#landing`.

      Memasukkan ini memungkinkan pengguna Anda untuk langsung masuk DataBrew.

   d. Pilih Lampirkan kebijakan AWS terkelola, cari DataBrew, dan pilih AwsGlueDataBrewFullAccessPolicy. Memilih ini memberi pengguna Anda semua izin yang mereka butuhkan. DataBrew Anda dapat menemukan detail lebih lanjut diMenambahkan kebijakan IAM untuk pengguna konsol.

   e. (Opsional) Pilih Buat kebijakan izin khusus dan sesuaikan izin untuk pengguna Anda.

5. Di panel navigasi Pusat Identitas IAM, pilih Grup, dan pilih Buat grup. Masukkan nama grup dan pilih Buat.

6. Tambahkan pengguna ke toko IAM Identity Center:

   a. Di panel navigasi Pusat Identitas IAM, pilih Pengguna.

   b. Pada layar Tambahkan pengguna, masukkan informasi yang diperlukan dan pilih Kirim email ke pengguna dengan instruksi pengaturan kata sandi. Pengguna harus mendapatkan email tentang langkah-langkah pengaturan berikutnya.

   c. Pilih Berikutnya: Grup, pilih grup yang Anda inginkan, dan pilih Tambah pengguna.

      Pengguna harus menerima email yang mengundang mereka untuk menggunakan SSO. Dalam email ini, mereka harus memilih Terima undangan dan mengatur kata sandi. Mereka juga dapat menemukan URL portal di email. Mereka dapat menggunakan URL ini untuk mengakses DataBrew.

7. Tetapkan setiap pengguna ke akun:

a. Buka konsol Pusat Identitas IAM, dan di panel navigasi, pilih AWS akun.

b. Pilih AWS organisasi dan pilih AWS akun.

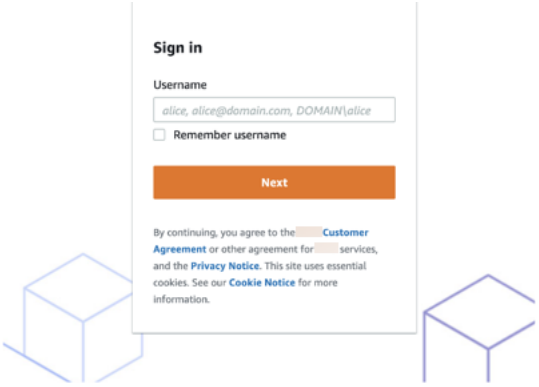
c. Pada layar Tetapkan Pengguna, pilih tab Grup dan pilih grup yang Anda inginkan.

d. Pilih Selanjutnya: Rangkaian izin.

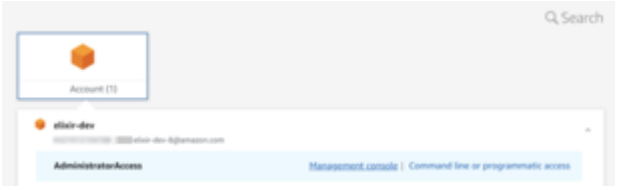
e. Pilih set izin untuk DataBrew, dan pilih Selesai.

Langkah-langkah login untuk pengguna IAM Identity Center-enabled

1. Masuk AWS menggunakan Center-enabled akun Identitas IAM.



2. Klik pada Identitas AWS akun



3. Klik Konsol manajemen untuk mengarahkan ulang satu klik ke konsol. DataBrew

Menggunakan DataBrew sebagai ekstensi di JupyterLab

> **Warning**
>
> AWS Glue DataBrew JupyterLab Dukungan ekstensi berakhir pada 31 Desember 2024 karena JupyterLab 3 akan mencapai akhir dukungan. Untuk informasi lebih lanjut, lihat JupyterLab 3 akhir pemeliharaan.

Jika Anda lebih suka menyiapkan data di lingkungan Jupyter Notebook, Anda dapat menggunakan semua kemampuan in.AWS Glue DataBrew JupyterLab

JupyterLab adalah lingkungan pengembangan interaktif berbasis web untuk Jupyter Notebook. Di JupyterLab halaman web lokal, Anda dapat menambahkan bagian untuk terminal, sesi SQL, Python, dan banyak lagi. Setelah menginstal AWS Glue DataBrew ekstensi, Anda dapat menambahkan bagian untuk DataBrew konsol. Ini berjalan dengan notebook yang ada atau ekstensi lain yang sudah Anda miliki, langsung dari JupyterLab lingkungan.

Topik

- Prasyarat
- Mengkonfigurasi JupyterLab untuk menggunakan ekstensi
- Mengaktifkan DataBrew ekstensi untuk JupyterLab

Prasyarat

Sebelum Anda mulai, atur item berikut:

- AWS Akun — Jika Anda belum memilikinya, mulailah dengan Menyiapkan yang baru AWS akun.
- Pengguna AWS Identity and Access Management(IAM) dengan akses ke izin yang diperlukan untuk DataBrew — Untuk informasi selengkapnya, lihat. Menambahkan pengguna atau grup dengan DataBrew izin
- Peran IAM untuk digunakan dalam DataBrew operasi — Anda dapat menggunakan default, jika `AwsGlueDataBrewDataAccessRole` dikonfigurasi. Untuk mengatur peran IAM tambahan, lihat Menambahkan peran IAM dengan izin sumber daya data.
- JupyterLab Instalasi (versi 2.2.6 atau lebih tinggi) - Untuk informasi lebih lanjut, lihat topik berikut dalam dokumentasi: JupyterLab
  - JupyterLab prasyarat
  - JupyterLab instalasi - Kami merekomendasikan penggunaan `pip install jupyterlab`.
- Node.js Instalasi (versi 12.0 atau lebih besar).
- Instalasi AWS Command Line Interface(AWS CLI) - Untuk informasi lebih lanjut, lihat Menyiapkan AWS CLI.
- Instalasi AWS proxy Jupyter (`pip install aws-jupyter-proxy`) — Ekstensi ini digunakan dengan titik akhir AWS layanan untuk meneruskan kredensyal Anda dengan aman.AWS Untuk informasi lebih lanjut, lihat aws-jupyter-proxy di. GitHub

Untuk memverifikasi bahwa Anda telah menginstal prasyarat, Anda dapat menjalankan pengujian yang mirip dengan yang berikut di baris perintah, seperti yang ditunjukkan pada contoh berikut.

```
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
```

Outputnya akan terlihat seperti berikut ini. Direktori bervariasi menurut sistem operasi dan konfigurasi.

```
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
```

Mengkonfigurasi JupyterLab untuk menggunakan ekstensi

Setelah Anda menginstal JupyterLab, Anda perlu mengkonfigurasinya untuk mengamankan akses data dan untuk mengaktifkan ekstensi server.

Untuk mengkonfigurasi kata sandi dan enkripsi

1. Tetapkan kata sandi untuk melindungi data yang Anda rencanakan untuk ditambahkan di ekstensi. Jupyter menyediakan utilitas kata sandi. Jalankan perintah berikut dan masukkan kata sandi pilihan Anda pada prompt.

   ```
   jupyter notebook password
   ```

   Output-nya akan terlihat seperti berikut.

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

2. Aktifkan enkripsi di server Jupyter. Jika Anda menginstal Jupyter di mesin lokal Anda, dan tidak ada yang dapat mengaksesnya melalui jaringan, Anda dapat melewati langkah ini.

   Untuk mengatur enkripsi dengan Transport Layer Security (TLS), buat sertifikat yang disesuaikan untuk lingkungan Anda. Untuk informasi lebih lanjut, Menggunakan Let's Encrypt dalam Mengamankan server dalam dokumentasi Jupyter.

3. Untuk memulai JupyterLab, jalankan perintah berikut di command prompt.

   ```
   jupyter lab
   ```

   Untuk informasi selengkapnya, lihat Memulai JupyterLab dalam JupyterLab dokumentasi.

4. Saat JupyterLab sedang berjalan, Anda dapat mengaksesnya di URL yang mirip dengan yang berikut ini: http://localhost:8888/lab. Jika Anda mengatur enkripsi, gunakan `https` sebagai penggantihttp. Jika Anda menyesuaikan port, ganti nomor port Anda alih-alih8888.

Gunakan prosedur berikut untuk mengaktifkan ekstensi pihak ketiga.

Untuk mengaktifkan ekstensi pihak ketiga di JupyterLab

1. Di JupyterLab halaman web, pilih ikon Extension Manager di menu di sebelah kiri.
2. Baca peringatan tentang risiko menjalankan ekstensi pihak ketiga. Hanya instal ekstensi dari pengembang yang Anda percayai.
3. Untuk mengaktifkan ekstensi pihak ketiga JupyterLab, pilih Aktifkan.

4. Ikuti petunjuk untuk membangun kembali dan memuat ulang. JupyterLab

Mengaktifkan DataBrew ekstensi untuk JupyterLab

Setelah Anda memiliki instalasi aman JupyterLab dengan ekstensi diaktifkan, instal DataBrew ekstensi sehingga Anda dapat menjalankan DataBrew di notebook Anda.

Untuk menginstal ekstensi untuk DataBrew (konsol)

1. Untuk memulai JupyterLab, jalankan perintah berikut di command prompt.

   ```
   jupyter lab
   ```

2. Di JupyterLab halaman web, pilih ikon Extension Manager di menu di sebelah kiri.
3. Cari DataBrew ekstensi dengan memasukkan "**`brew`**" untuk Cari di kiri atas.
4. Temukan aws_glue_databrew_jupyter dalam daftar, tetapi jangan klik. Jika Anda mengklik nama ekstensi yang disorot, jendela browser baru terbuka dengan halaman aws_glue_databrew_jupyter aktif. GitHub
5. Untuk menginstal DataBrew ekstensi, pilih salah satu dari yang berikut ini:

   - Di baris perintah, jalankan`jupyter labextension install aws_glue_databrew_jupyter`.
   - Pilih Instal di bagian bawah kartu ekstensi, di bawah "aws_glue_databrew_jupyter" dengan huruf abu-abu.

   DataBrew ekstensi kompatibel dengan JupyterLab versi 1.2 dan 2.x.
6. Untuk memverifikasi bahwa itu diinstal, jalankan`jupyter labextension list`. Outputnya akan terlihat seperti berikut ini.

   ```
   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. Membangun kembali JupyterLab dengan menggunakan salah satu dari berikut ini:

   - Pada prompt perintah, jalankan`jupyter lab build`.
   - Di halaman web, pilih Rebuild di kiri atas.

8. Saat build selesai, lakukan salah satu hal berikut:

   - Pada prompt perintah, jalankan`jupyter lab`.
   - Di halaman web, pilih Muat ulang pada pesan Build Complete.

9. Di JupyterLab halaman web, tutup Manajer Ekstensi dengan memilih ikonnya di menu di sebelah kiri.

   Untuk membuka ekstensi, pilih Luncurkan AWS Glue DataBrew dari bagian Lainnya pada tab Peluncur. Ekstensi menggunakan AWS CLI konfigurasi Anda saat ini untuk kunci akses dan pengaturan AWS wilayah.

Setelah Anda menyelesaikan pengaturan, Anda dapat menggunakan AWS Glue DataBrewtab untuk berinteraksi dengan DataBrew dari dalam JupyterLab.

Memulai dengan AWS Glue DataBrew

Anda dapat menggunakan tutorial berikut untuk memandu Anda dalam membuat DataBrew proyek pertama Anda. Anda memuat kumpulan data sampel, menjalankan transformasi pada kumpulan data tersebut, membuat resep untuk menangkap transformasi tersebut, dan menjalankan pekerjaan untuk menulis data yang diubah ke Amazon S3.

Topik

- Prasyarat
- Langkah 1: Buat proyek
- Langkah 2: Ringkas data
- Langkah 3: Tambahkan lebih banyak transformasi
- Langkah 4: Tinjau DataBrew sumber daya Anda
- Langkah 5: Buat profil data
- Langkah 6: Ubah kumpulan data
- Langkah 7: (Opsional) Bersihkan

Prasyarat

Sebelum Anda melanjutkan, ikuti instruksi yang berlaku diMenyiapkan AWS Glue DataBrew. Kemudian lanjutkan keLangkah 1: Buat proyek.

Langkah 1: Buat proyek

Pada langkah ini, Anda menggunakan DataBrew konsol untuk memulai proyek sampel dengan cepat.

Untuk membuat proyek

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol di https://console.aws.amazon.com/databrew/.
2. Pastikan AWS Wilayah Anda dipilih di kanan atas pada konsol. DataBrew Untuk daftar AWS Wilayah yang didukung oleh DataBrew, lihat DataBrew titik akhir dan kuota di.Referensi Umum AWS
3. Pada panel navigasi, pilih Projects, lalu pilih Create project.

4. Pada panel Detail proyek, lakukan hal berikut:

   - Untuk nama Proyek, masukkan`chess-project`.
   - Untuk resep Terlampir, buat resep baru. Nama yang disarankan untuk resep disediakan (`chess-project-recipe`).

5. Pada panel Pilih kumpulan data, pilih Contoh file.
6. Pada panel File sampel, pilih Gerakan permainan catur terkenal. Dataset ini berisi informasi terperinci tentang lebih dari 20.000 permainan catur.

   Untuk nama Dataset nama yang disarankan untuk kumpulan data disediakan ()`chess-games`.
7. Pada panel Izin akses, pilih. `AwsGlueDataBrewDataAccessRole` Ini adalah peran terkait layanan yang memungkinkan DataBrew akses bucket Amazon S3 Anda atas nama Anda.
8. Pilih Buat proyek, dan tunggu sampai DataBrew selesai mempersiapkan proyek. Jendela terlihat mirip dengan yang berikut ini.

   Data yang Anda lihat mewakili sampel dari `chess-games` kumpulan data. Secara default, sampel terdiri dari 500 baris pertama dari kumpulan data. Anda dapat mengubah pengaturan proyek ini nanti.

   Toolbar menyediakan akses ke ratusan transformasi data yang dapat Anda terapkan ke data.

   Panel resep di sebelah kanan di DataBrew konsol melacak transformasi yang Anda terapkan sejauh ini.

Langkah 2: Ringkas data

Pada langkah ini, Anda membuat DataBrew resep—serangkaian transformasi yang dapat diterapkan pada kumpulan data ini dan yang lainnya seperti itu. Ketika resep selesai, Anda mempublikasikannya sehingga tersedia untuk digunakan.

Dalam permainan catur, pemain dapat dinilai berdasarkan seberapa baik kinerja mereka melawan pemain lain. (Untuk informasi lebih lanjut, lihathttps://en.wikipedia.org/wiki/Chess_rating_system). Untuk tutorial ini, Anda hanya fokus pada permainan di mana kedua pemain adalah Kelas A, yang berarti bahwa peringkat mereka adalah 1800 atau lebih.

Untuk meringkas data

1. Pada bilah alat transformasi, pilih Filter, Berdasarkan Kondisi, Lebih besar dari atau sama dengan.
2. Tetapkan opsi ini sebagai berikut:

   - Kolom sumber - `white_rating`
   - Kondisi filter - Lebih besar dari atau sama dengan 1800

   Untuk melihat cara kerja transformasi, pilih Pratinjau perubahan. Lalu, pilih Terapkan.
3. Ulangi langkah sebelumnya, tetapi kali ini atur kolom Sumber ke`black_rating`. Setelah Anda menerapkan perubahan Anda, data sampel hanya berisi game-game di mana pemain di setiap sisi (hitam dan putih) adalah Kelas A atau lebih tinggi.
4. Ringkas data untuk menentukan berapa banyak game yang dimenangkan oleh masing-masing pihak. Untuk melakukan ini, pada bilah alat transformasi, pilih Grup.
5. Untuk properti Grup, lakukan hal berikut:

   a. Di baris pertama, pilih `winner` nama Kolom. Biarkan Agregat diatur ke Grup oleh.

   b. Di baris kedua, pilih `victory_status` nama Kolom. Biarkan Agregat diatur ke Grup oleh.

   c. Pilih Tambahkan kolom lain.

   d. Di baris ketiga, pilih `winner` nama Kolom. Atur Agregat ke Hitung.

   e. Untuk tipe Grup, pilih Grup sebagai tabel baru. Panel pratinjau menunjukkan kepada Anda seperti apa hasilnya.

   f. Pilih Selesai.
6. Pilih Publikasikan untuk menyimpan pekerjaan Anda, tepat di panel resep.
7. Untuk Deskripsi Versi, masukkan Versi pertama resep saya. Kemudian pilih Publikasikan.

Langkah 3: Tambahkan lebih banyak transformasi

Pada langkah ini, Anda menambahkan lebih banyak transformasi ke resep Anda dan menerbitkan versi lain darinya. Untuk menyempurnakan contoh kami, kami menggunakan informasi bahwa tidak semua permainan catur menghasilkan pemenang yang jelas; beberapa permainan dimainkan untuk hasil imbang.

Untuk menambahkan lebih banyak transformasi resep dan menerbitkan ulang

1. Dari bilah alat transformasi, pilih Filter, Berdasarkan Kondisi, Bukan untuk menghapus game yang dimainkan untuk hasil imbang.
2. Tetapkan opsi ini sebagai berikut:

   - Kolom sumber - `victory_status`
   - Kondisi filter - Tidak `draw`

   Untuk menambahkan transformasi ini ke resep Anda, pilih Terapkan.
3. Ubah data `victory_status` agar lebih bermakna. Untuk melakukan ini, dari bilah alat transformasi pilih Bersihkan, Ganti, Ganti nilai atau pola.
4. Tetapkan opsi ini sebagai berikut:

   - Kolom sumber - `victory_status`
   - Tentukan nilai yang akan diganti - Nilai atau pola
   - Nilai yang akan diganti - `mate`
   - Ganti dengan nilai - `checkmate`

   Untuk menambahkan transformasi ini ke resep Anda, pilih Terapkan.
5. Ulangi langkah sebelumnya, tetapi ubah `resign` ke`other player resigned`.
6. Ulangi langkah sebelumnya, tetapi ubah `outoftime` ke`time ran out`.
7. Pilih Publikasikan untuk menyimpan pekerjaan Anda, tepat di panel resep.

Langkah 4: Tinjau DataBrew sumber daya Anda

Sekarang setelah Anda bekerja dengan proyek sampel, tinjau DataBrew sumber daya yang Anda buat sejauh ini.

Untuk meninjau DataBrew sumber daya Anda

1. Pada panel navigasi, pilih Datasets.

   Saat Anda membuat proyek sampel, DataBrew buat kumpulan data untuk Anda (`chess-games`). File data sumber disimpan di Amazon S3, dan dalam format Microsoft Excel ()`chess-`

`games.xlsx`. File ini berisi metadata dari lebih dari 20.000 permainan catur. `chess-games`Dataset menyediakan informasi yang DataBrew perlu membaca data dalam file itu.

2. Pada panel navigasi, pilih Proyek.

   Anda akan melihat proyek yang Anda kerjakan di langkah sebelumnya (`chess-project`). Setiap proyek membutuhkan dataset, dalam hal `chess-games` ini. Setiap proyek juga memerlukan resep, sehingga Anda dapat menambahkan langkah-langkah transformasi data saat Anda berjalan. Saat Anda membuat proyek sampel ini, DataBrew buat resep baru (kosong) untuk Anda, dan lampirkan ke proyek.

3. Pada panel navigasi, pilih Resep, dan di kolom Nama resep, pilih catur-proyek-resep. Ini menunjukkan kepada Anda resep yang DataBrew dibuat untuk proyek Anda, dan bahwa Anda telah menyempurnakan dengan menambahkan langkah-langkah transformasi ke dalamnya.

4. Di sebelah kiri, lihat versi resep yang telah diterbitkan. Pilih salah satunya untuk melihat tab Langkah Resep, yang menunjukkan detail resep dan langkah-langkah untuk versi itu.

5. Lihat tab Garis keturunan data, yang menunjukkan dari mana data berasal dan bagaimana data itu digunakan. Untuk lebih jelasnya, pilih salah satu ikon dalam diagram.

Langkah 5: Buat profil data

Saat Anda bekerja dengan proyek, DataBrew menampilkan statistik seperti jumlah baris dalam sampel dan distribusi nilai unik di setiap kolom. Statistik ini, dan banyak lagi, mewakili profil sampel.

Untuk meminta profil data, buat dan jalankan pekerjaan profil.

Untuk membuat profil kumpulan data

1. Pada panel navigasi, pilih Jobs.
2. Pada tab Profile jobs, pilih Create job.
3. Untuk nama Job, masukkan`chess-data-profile`.
4. Untuk jenis Job, pilih Buat pekerjaan profil.
5. Pada panel input Job, lakukan hal berikut:

   - Untuk Run on, pilih Dataset.
   - Pilih Pilih kumpulan data untuk melihat daftar kumpulan data yang tersedia, lalu pilih. `chess-games`

6. Pada panel pengaturan keluaran Job, lakukan hal berikut:

   - Untuk jenis File, pilih JSON (JavaScript Object Notation).
   - Pilih lokasi S3 untuk melihat daftar bucket Amazon S3 yang tersedia, dan pilih bucket yang akan digunakan. Kemudian pilih Browse. Dalam daftar folder, pilih`databrew-output`, dan pilih Pilih.
7. Pada panel Izin akses, pilih. `AwsGlueDataBrewDataAccessRole` Ini adalah peran terkait layanan yang memungkinkan DataBrew akses bucket Amazon S3 Anda atas nama Anda.
8. Pilih Buat dan jalankan pekerjaan. DataBrew membuat pekerjaan dengan pengaturan Anda, dan kemudian menjalankannya.
9. Pada panel Riwayat Job run, tunggu status lowongan berubah dari `Running` menjadi`Succeeded`.
10. Untuk melihat profil, pilih LIHAT PROFIL:

    Jendela DATASETS ditampilkan. Luangkan waktu untuk menjelajahi tab berikut:

    - Pratinjau kumpulan data
    - Ikhtisar profil
    - Statistik kolom
    - Statistik garis keturunan data

Langkah 6: Ubah kumpulan data

Sampai sekarang, Anda menguji resep Anda hanya pada sampel kumpulan data. Sekarang saatnya mengubah seluruh kumpulan data dengan membuat pekerjaan DataBrew resep.

Saat pekerjaan berjalan, DataBrew terapkan resep Anda ke semua data dalam kumpulan data, dan tulis data yang diubah ke bucket Amazon S3. Data yang ditransformasikan terpisah dari dataset asli. DataBrew tidak mengubah sumber data.

Sebelum melanjutkan, pastikan Anda memiliki bucket Amazon S3 di akun yang dapat Anda tulis. Dalam bucket itu, buat folder untuk menangkap output pekerjaan dari DataBrew. Untuk melakukan langkah-langkah ini, gunakan prosedur berikut.

Untuk membuat bucket dan folder S3 untuk menangkap output pekerjaan

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. https://console.aws.amazon.com/databrew/

   Jika Anda sudah memiliki bucket Amazon S3 yang tersedia, dan Anda memiliki izin menulis untuk itu, lewati langkah berikutnya.
2. Jika Anda tidak memiliki bucket Amazon S3, pilih Buat ember. Untuk nama Bucket, masukkan nama unik untuk bucket baru Anda. Pilih Buat bucket.
3. Dari daftar ember, pilih salah satu yang ingin Anda gunakan.
4. Pilih Buat folder.
5. Untuk nama Folder, masukkan`databrew-output`, dan pilih Buat folder.

Setelah Anda membuat bucket dan folder Amazon S3 untuk memuat pekerjaan, jalankan pekerjaan Anda dengan menggunakan prosedur berikut.

Untuk membuat dan menjalankan pekerjaan resep

1. Pada panel navigasi, pilih Jobs.
2. Pada tab Recipe jobs, pilih Create job.
3. Untuk nama Job, masukkan`chess-winner-summary`.
4. Untuk jenis Job, pilih Create a recipe job.
5. Pada panel input Job, lakukan hal berikut:
   - Untuk Run on, pilih Dataset.
   - Pilih Pilih kumpulan data untuk melihat daftar kumpulan data yang tersedia, lalu pilih. `chess-games`
   - Pilih Pilih resep untuk melihat daftar resep yang tersedia, dan pilih`chess-project-recipe`.
6. Pada panel pengaturan keluaran Job, lakukan hal berikut:
   - Jenis file - pilih CSV (nilai yang dipisahkan koma).
   - Lokasi S3 - pilih bidang ini untuk melihat daftar bucket Amazon S3 yang tersedia, dan pilih bucket yang akan digunakan. Kemudian pilih Browse. Dalam daftar folder, pilih`databrew-output`, dan pilih Pilih.

7. Pada panel Izin akses, pilih. `AwsGlueDataBrewDataAccessRole` Peran terkait layanan ini memungkinkan DataBrew akses bucket Amazon S3 Anda atas nama Anda.
8. Pilih Buat dan jalankan pekerjaan. DataBrew membuat pekerjaan dengan pengaturan Anda, dan kemudian menjalankannya.
9. Pada panel Riwayat Job run, tunggu status lowongan berubah dari `Running` menjadi`Succeeded`.
10. Pilih Output untuk mengakses konsol Amazon S3. Pilih bucket S3 Anda, lalu pilih `databrew-output` folder untuk mengakses output pekerjaan.
11. (Opsional) Pilih Unduh untuk mengunduh file dan melihat isinya.

Langkah 7: (Opsional) Bersihkan

Walkthrough selesai. Anda dapat tetap menggunakan sumber daya Amazon S3 DataBrew dan Amazon yang Anda buat, atau menghapusnya.

Untuk membersihkan sumber daya

1. Buka DataBrew konsol di https://console.aws.amazon.com/databrew/, dan pada panel navigasi, pilih Proyek.
2. Pilih proyek Anda (Contoh proyek). Untuk Tindakan, pilih Hapus.
3. Pada panel Delete Sample project, pilih Hapus resep terlampir. Lalu pilih Hapus. Proyek Anda, bersama dengan resep dan pekerjaannya, akan dihapus.
4. Pada panel navigasi, pilih Datasets.
5. Pilih kumpulan data Anda (`chess-games`), dan untuk Tindakan, pilih Hapus.
6. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/ Hapus `databrew-output` folder dan isinya.

   (Opsional) Jika Anda yakin tidak lagi membutuhkan bucket Amazon S3, Anda dapat menghapusnya.

Menghubungkan ke data dengan AWS Glue DataBrew

Dalam AWS Glue DataBrew, kumpulan data mewakili data yang diunggah dari file atau disimpan di tempat lain. Misalnya, data dapat disimpan di Amazon S3, di sumber data JDBC yang didukung, atau Katalog Data.AWS Glue Jika Anda tidak mengunggah file secara langsung DataBrew, kumpulan data juga berisi detail tentang cara DataBrew menghubungkan ke data.

Saat Anda membuat kumpulan data (misalnya,`inventory-dataset`), Anda memasukkan detail koneksi hanya sekali. Dari titik itu, DataBrew dapat mengakses data yang mendasarinya untuk Anda. Dengan pendekatan ini, Anda dapat membuat proyek dan mengembangkan transformasi untuk data Anda, tanpa harus khawatir tentang detail koneksi atau format file.

Topik

- Jenis file yang didukung untuk sumber data
- Koneksi yang didukung untuk sumber data dan output
- Menggunakan dataset di AWS Glue DataBrew
- Menghubungkan ke data Anda
- Menghubungkan ke data dalam file teks dengan DataBrew
- Menghubungkan data dalam beberapa file di Amazon S3
- Jenis Data
- Tipe data tingkat lanjut

Jenis file yang didukung untuk sumber data

Persyaratan file berikut berlaku untuk file yang disimpan di Amazon S3 dan file yang Anda unggah dari drive lokal. DataBrew mendukung format file berikut: nilai dipisahkan koma (CSV), Microsoft Excel, JSON, ORC, dan Parket. Anda dapat menggunakan file dengan ekstensi yang tidak standar atau tanpa ekstensi jika file tersebut dari salah satu jenis yang didukung.

Jika DataBrew tidak dapat menyimpulkan jenis file, pastikan untuk memilih sendiri jenis file yang benar (CSV, Excel, JSON, ORC, atau Parket). File CSV, JSON, ORC, dan Parquet terkompresi didukung, tetapi file CSV dan JSON harus menyertakan codec kompresi sebagai ekstensi file. Jika Anda mengimpor folder, semua file dalam folder harus dari jenis file yang sama.

Format file dan algoritma kompresi yang didukung ditunjukkan pada tabel berikut.

> **Note**
>
> File CSV, Excel, dan JSON harus dikodekan dengan Unicode (). UTF-8

Format	Ekstensi file (opsional)	Ekstensi untuk file terkompresi (wajib)

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

Buku kerja Microsoft Excel	`.xlsx`	Tidak ada dukungan kompresi

JSON (dokumen JSON dan baris JSON)	`.json, .jsonl`	`.gz`
		`.snappy`
		`.lz4`
		`.bz2`
		`.deflate`

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

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

Koneksi yang didukung untuk sumber data dan output

Anda dapat terhubung ke sumber data berikut untuk pekerjaan DataBrew resep. Ini termasuk sumber data apa pun yang bukan file yang Anda unggah langsung. DataBrew Sumber data yang Anda gunakan mungkin disebut database, gudang data, atau sesuatu yang lain. Kami menyebut semua penyedia data sebagai sumber data atau koneksi.

Anda dapat membuat kumpulan data menggunakan salah satu dari berikut ini sebagai sumber data.

Anda juga dapat menggunakan database Amazon S3,AWS Glue Data Catalog, atau JDBC yang didukung melalui Amazon RDS untuk output pekerjaan resep. DataBrew Amazon AppFlow dan AWS Data Exchange tidak didukung penyimpanan data untuk output pekerjaan DataBrew resep.

- Amazon S3

  Anda dapat menggunakan S3 untuk menyimpan dan melindungi sejumlah data. Untuk membuat kumpulan data, Anda menentukan URL S3 tempat DataBrew dapat mengakses file data, misalnya: `s3://your-bucket-name/inventory-data.csv`

  DataBrew juga dapat membaca semua file dalam folder S3, yang berarti bahwa Anda dapat membuat dataset yang mencakup beberapa file. Untuk melakukan ini, tentukan URL S3 dalam formulir ini:`s3://your-bucket-name/your-folder-name/`.

  DataBrew hanya mendukung kelas penyimpanan Amazon S3 berikut: Standar, Redundansi yang Dikurangi, dan S3 One Standard-IA. Zone-IA DataBrew mengabaikan file dengan kelas penyimpanan lainnya. DataBrew juga mengabaikan file kosong (file yang berisi 0 byte). Untuk informasi selengkapnya tentang kelas penyimpanan Amazon S3, lihat Menggunakan kelas penyimpanan Amazon S3 di Panduan Pengguna Konsol Amazon S3.
- AWS Glue Data Catalog

  Anda dapat menggunakan Katalog Data untuk menentukan referensi ke data yang disimpan di AWS Cloud. Dengan Katalog Data, Anda dapat membangun koneksi ke tabel individual dalam layanan berikut:

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

DataBrew juga dapat membaca semua file di folder Amazon S3, yang berarti Anda dapat membuat kumpulan data yang mencakup beberapa file. Untuk melakukan ini, tentukan URL Amazon S3 dalam formulir ini: `s3://your-bucket-name/your-folder-name/`

Untuk digunakan DataBrew, tabel Amazon S3 yang didefinisikan dalam AWS Glue Data Catalog, harus memiliki properti tabel yang ditambahkan ke tabel yang disebut `aclassification`, yang mengidentifikasi format data sebagai`csv`, atau `jsonparquet`, dan as. `typeOfData` `file` Jika properti tabel tidak ditambahkan saat tabel dibuat, Anda dapat menambahkannya menggunakan AWS Glue konsol.

DataBrew hanya mendukung kelas penyimpanan Amazon S3 Standar, Redundansi yang Dikurangi, dan S3 One Standard-IA. Zone-IA DataBrew mengabaikan file dengan kelas penyimpanan lainnya. DataBrew juga mengabaikan file kosong (file yang berisi 0 byte). Untuk informasi selengkapnya tentang kelas penyimpanan Amazon S3, lihat Menggunakan kelas penyimpanan Amazon S3 di Panduan Pengguna Konsol Amazon S3.

DataBrew juga dapat mengakses tabel AWS Glue Data Catalog S3 dari akun lain jika kebijakan sumber daya yang sesuai dibuat. Anda dapat membuat kebijakan di AWS Glue konsol pada tab Pengaturan di bawah Katalog Data. Berikut ini adalah contoh kebijakan khusus untuk satu Wilayah AWS.

> **Warning**
>
> Ini adalah kebijakan sumber daya yang sangat permisif yang memberikan akses *`$ACCOUNT_TO`* tak terbatas ke Katalog Data. *`$ACCOUNT_FROM`* Dalam kebanyakan kasus, kami menyarankan Anda mengunci kebijakan sumber daya Anda ke katalog atau tabel tertentu. Untuk informasi selengkapnya, lihat kebijakan AWS Glue sumber daya untuk kontrol akses di Panduan AWS Glue Pengembang.

Dalam beberapa kasus, Anda mungkin ingin membuat proyek atau menjalankan pekerjaan *`$ACCOUNT_TO`* dengan tabel AWS Glue Data Catalog S3 *`$ACCOUNT_FROM`* yang mengarah ke lokasi S3 yang juga ada.AWS Glue DataBrew*`$ACCOUNT_FROM`* Dalam kasus seperti itu, peran IAM yang digunakan saat membuat proyek dan pekerjaan *`$ACCOUNT_TO`* harus memiliki izin untuk membuat daftar dan mendapatkan objek di lokasi S3 itu dari. *`$ACCOUNT_FROM`* Untuk informasi selengkapnya, lihat Memberikan akses lintas akun di Panduan AWS Glue Pengembang.

- Data terhubung menggunakan driver JDBC

Anda dapat membuat kumpulan data dengan menghubungkan ke data dengan driver JDBC yang didukung. Untuk informasi selengkapnya, lihat Menggunakan driver dengan AWS Glue DataBrew.

DataBrew secara resmi mendukung sumber data berikut menggunakan Java Database Connectivity (JDBC):

- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Amazon Redshift
- Konektor Kepingan Salju untuk Spark

Sumber data dapat ditemukan di mana saja Anda dapat terhubung dengannya DataBrew. Daftar ini hanya mencakup koneksi JDBC yang telah kami uji dan karenanya dapat mendukung.

Amazon Redshift dan Snowflake Connector untuk sumber data Spark dapat dihubungkan dengan salah satu cara berikut:

- Dengan nama tabel.
- Dengan query SQL yang mencakup beberapa tabel dan operasi.

Kueri SQL dijalankan ketika Anda memulai proyek atau menjalankan pekerjaan.

Untuk terhubung ke data yang memerlukan driver JDBC yang tidak terdaftar, pastikan driver tersebut kompatibel dengan JDK 8. Untuk menggunakan driver, simpan di S3 dalam ember tempat Anda dapat mengaksesnya dengan peran IAM Anda. DataBrew Kemudian arahkan dataset Anda ke file driver. Untuk informasi selengkapnya, lihat Menggunakan driver dengan AWS Glue DataBrew.

Contoh kueri untuk SQL-based kumpulan data:

```
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
```

Keterbatasan SQL Kustom

Jika Anda menggunakan koneksi JDBC untuk mengakses data untuk DataBrew kumpulan data, ingatlah hal berikut:

- AWS Glue DataBrew tidak memvalidasi SQL kustom yang Anda berikan sebagai bagian dari pembuatan dataset. Query SQL akan dieksekusi ketika Anda memulai proyek atau menjalankan pekerjaan. DataBrew mengambil kueri yang Anda berikan dan meneruskannya ke mesin database menggunakan driver JDBC default atau yang disediakan.
- Dataset yang dibuat dengan kueri yang tidak valid akan gagal saat digunakan dalam proyek atau pekerjaan. Validasi kueri Anda sebelum membuat kumpulan data.
- Fitur Validasi SQL hanya tersedia untuk sumber Redshift-based data Amazon.
- Jika Anda ingin menggunakan kumpulan data dalam proyek, batasi runtime kueri SQL hingga kurang dari tiga menit untuk menghindari batas waktu selama pemuatan proyek. Periksa runtime kueri sebelum membuat proyek.

- Amazon AppFlow

  Menggunakan Amazon AppFlow, Anda dapat mentransfer data ke Amazon S3 dari aplikasi pihak ketiga (Software-as-a-Service SaaS) seperti Salesforce, Zendesk, Slack, dan. ServiceNow Anda kemudian dapat menggunakan data untuk membuat DataBrew dataset.

  Di Amazon AppFlow, Anda membuat koneksi dan alur untuk mentransfer data antara aplikasi pihak ketiga Anda dan aplikasi tujuan. Saat menggunakan Amazon AppFlow dengan DataBrew, pastikan bahwa aplikasi AppFlow tujuan Amazon adalah Amazon S3. Aplikasi AppFlow tujuan Amazon selain Amazon S3 tidak muncul di konsol. DataBrew Untuk informasi selengkapnya tentang mentransfer data dari aplikasi pihak ketiga Anda serta membuat AppFlow koneksi dan alur Amazon, lihat AppFlow dokumentasi Amazon.

  Bila Anda memilih Connect new dataset di tab Datasets dan DataBrew klik Amazon AppFlow, Anda akan melihat semua flow di Amazon AppFlow yang dikonfigurasi dengan Amazon S3 sebagai aplikasi tujuan. Untuk menggunakan data flow untuk kumpulan data Anda, pilih alur tersebut.

  Memilih Buat alur, Kelola alur, dan Lihat detail untuk Amazon AppFlow di DataBrew konsol akan membuka AppFlow konsol Amazon sehingga Anda dapat melakukan tugas tersebut.

Setelah membuat kumpulan data dari Amazon AppFlow, Anda dapat menjalankan alur dan melihat detail proses alur terbaru saat melihat detail kumpulan data atau detail pekerjaan. Saat Anda menjalankan alur DataBrew, kumpulan data diperbarui di S3 dan siap digunakan. DataBrew

Situasi berikut dapat muncul saat Anda memilih AppFlow aliran Amazon di DataBrew konsol untuk membuat kumpulan data:

- Data belum dikumpulkan - Jika pemicu aliran Jalankan sesuai permintaan atau Jalankan sesuai jadwal dengan transfer data lengkap, pastikan untuk menggabungkan data untuk alur sebelum menggunakannya untuk membuat DataBrew kumpulan data. Mengagregasi aliran menggabungkan semua catatan dalam aliran ke dalam satu file. Alur dengan tipe pemicu Jalankan sesuai jadwal dengan transfer data tambahan, atau Run on event tidak memerlukan agregasi. Untuk menggabungkan data di Amazon AppFlow, pilih Edit konfigurasi alur **>** Detail tujuan **>** Pengaturan tambahan **>** Preferensi transfer data.
- Flow belum dijalankan - Jika status run untuk aliran kosong, itu berarti salah satu dari berikut ini:
  - Jika pemicu untuk menjalankan aliran adalah Run on demand, aliran belum dijalankan.
  - Jika pemicu untuk menjalankan alur adalah Run on event, peristiwa pemicu belum terjadi.
  - Jika pemicu untuk menjalankan alur adalah Jalankan sesuai jadwal, proses terjadwal belum terjadi.

  Sebelum membuat dataset dengan flow, pilih Run flow untuk flow tersebut.

  Untuk informasi selengkapnya, lihat AppFlow Aliran Amazon di Panduan AppFlow Pengguna Amazon.
- AWS Data Exchange

  Anda dapat memilih dari ratusan sumber data pihak ketiga yang tersedia di AWS Data Exchange. Dengan berlangganan sumber data ini, Anda mendapatkan versi data terbaru.

  Untuk membuat kumpulan data, Anda menentukan nama produk AWS Data Exchange data yang Anda berlangganan dan berhak untuk digunakan.

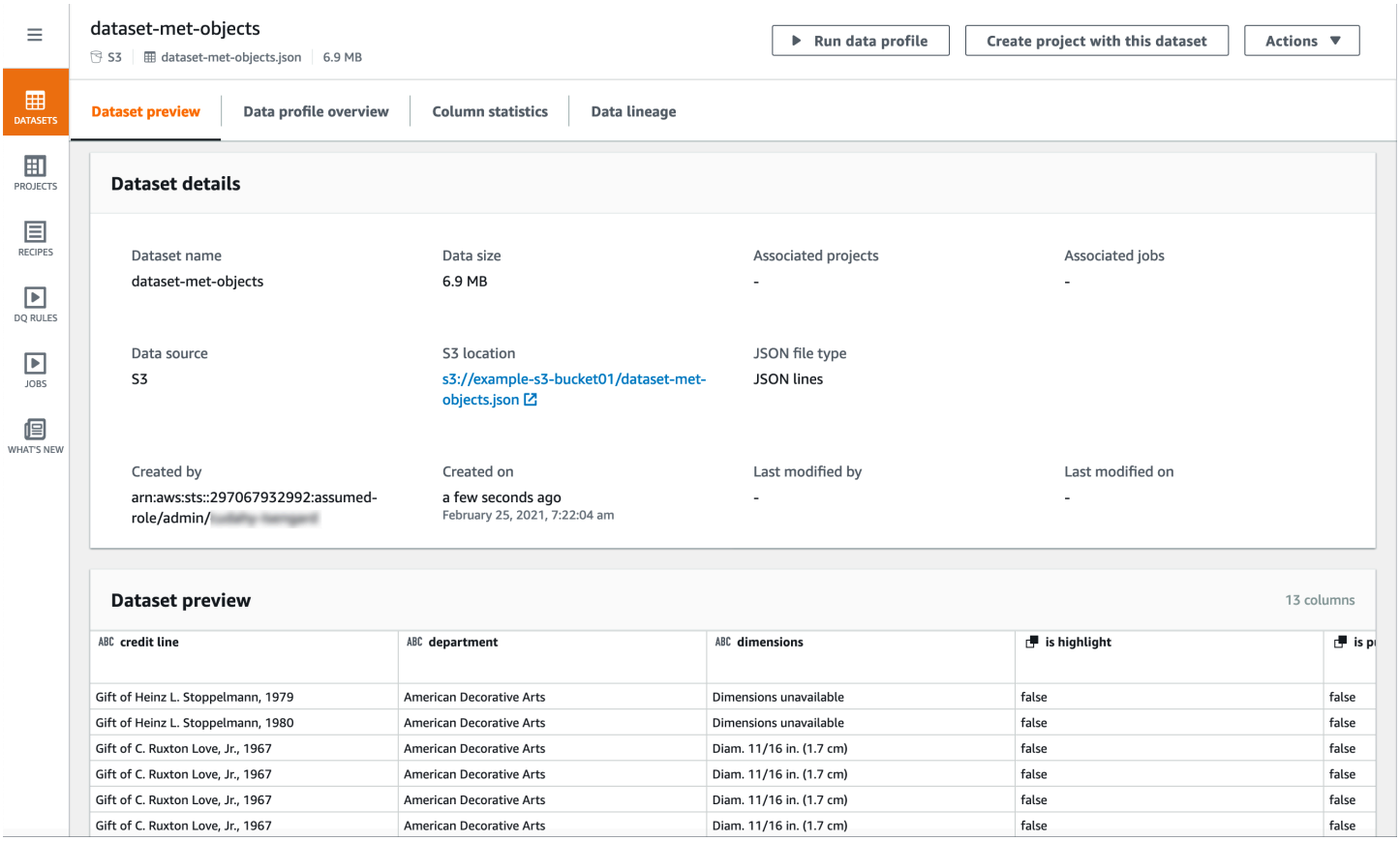
Menggunakan dataset di AWS Glue DataBrew

Untuk melihat daftar kumpulan data Anda di DataBrew konsol, pilih DATASET di sebelah kiri. Di halaman kumpulan data, Anda dapat melihat informasi terperinci untuk setiap kumpulan data dengan mengklik namanya atau memilih Tindakan, Edit dari menu konteksnya.

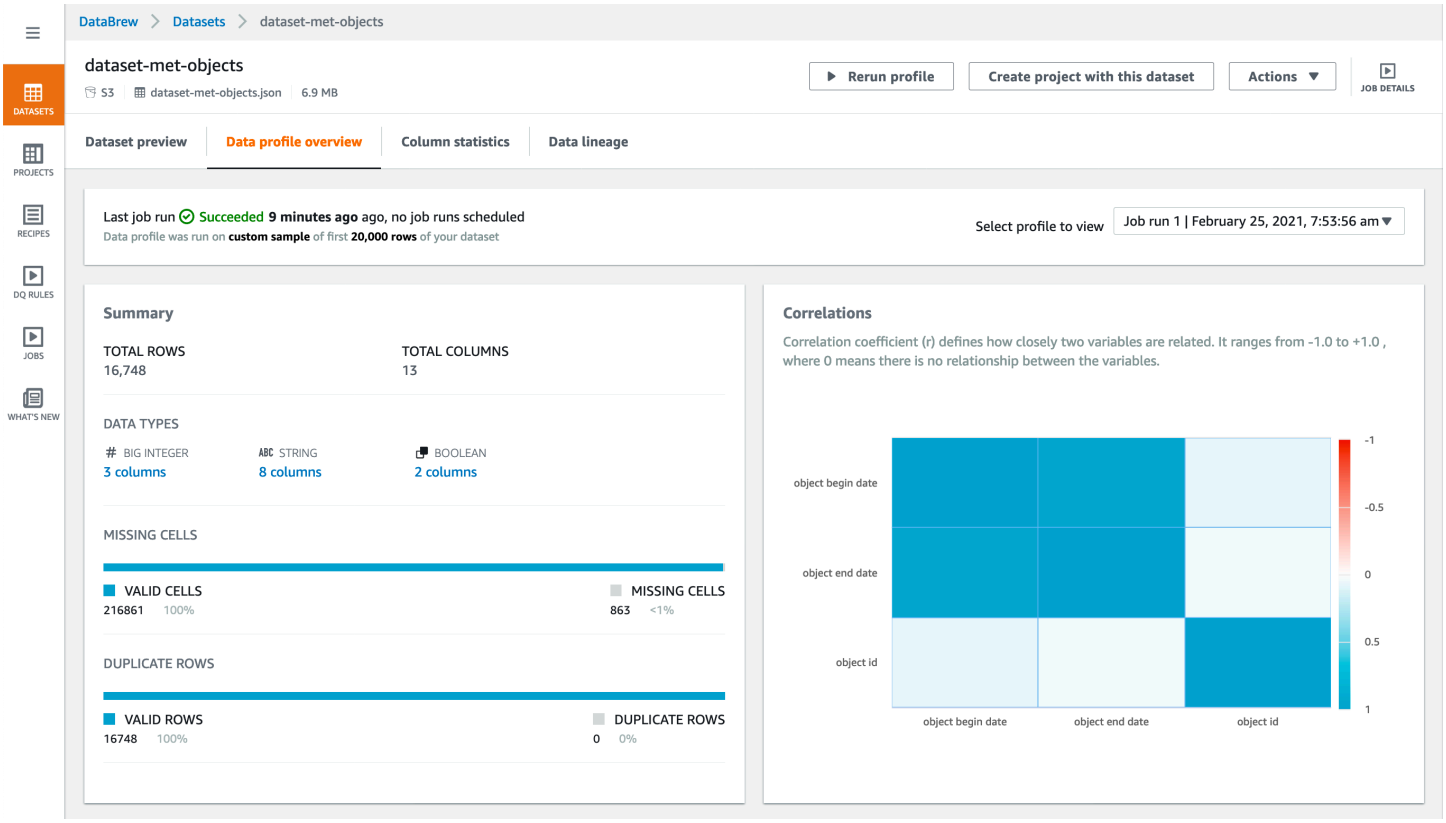
Untuk membuat kumpulan data baru, Anda memilih DATASET, Connect new dataset. Sumber data yang berbeda memiliki parameter koneksi yang berbeda, dan Anda memasukkannya sehingga DataBrew dapat terhubung. Saat Anda menyimpan koneksi dan memilih Buat kumpulan data, DataBrew sambungkan ke data Anda dan mulai memuat data. Untuk informasi selengkapnya, lihat Menghubungkan ke data Anda.

Halaman dataset memiliki elemen berikut untuk membantu Anda menjelajahi data Anda.

Pratinjau kumpulan data - Pada tab ini, Anda dapat menemukan informasi koneksi untuk kumpulan data dan ikhtisar keseluruhan struktur kumpulan data, seperti yang ditunjukkan berikut.

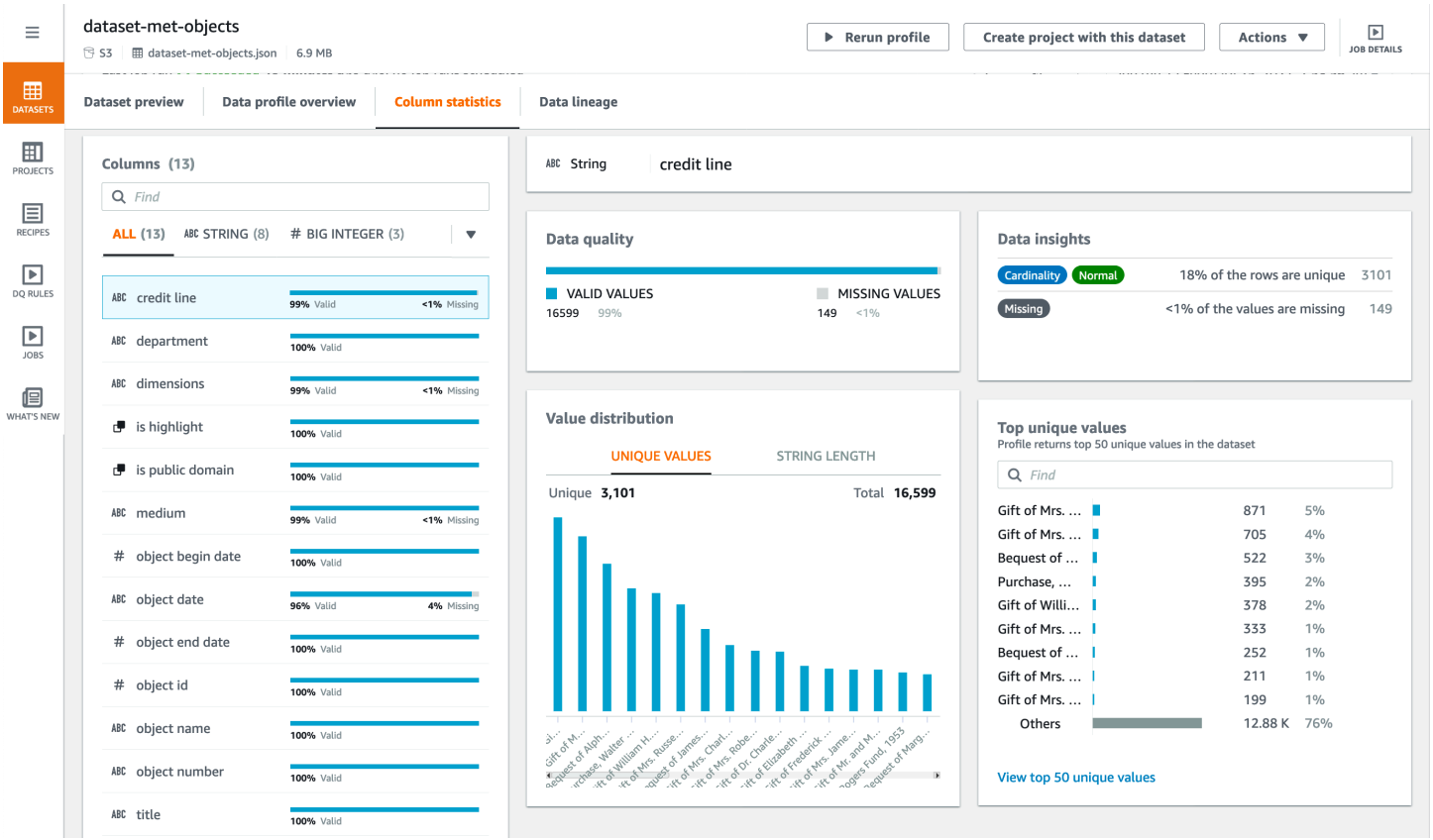


Ikhtisar profil data — Pada tab ini, Anda dapat menemukan profil data grafis statistik dan volumetrik untuk kumpulan data Anda, seperti yang ditunjukkan berikut.

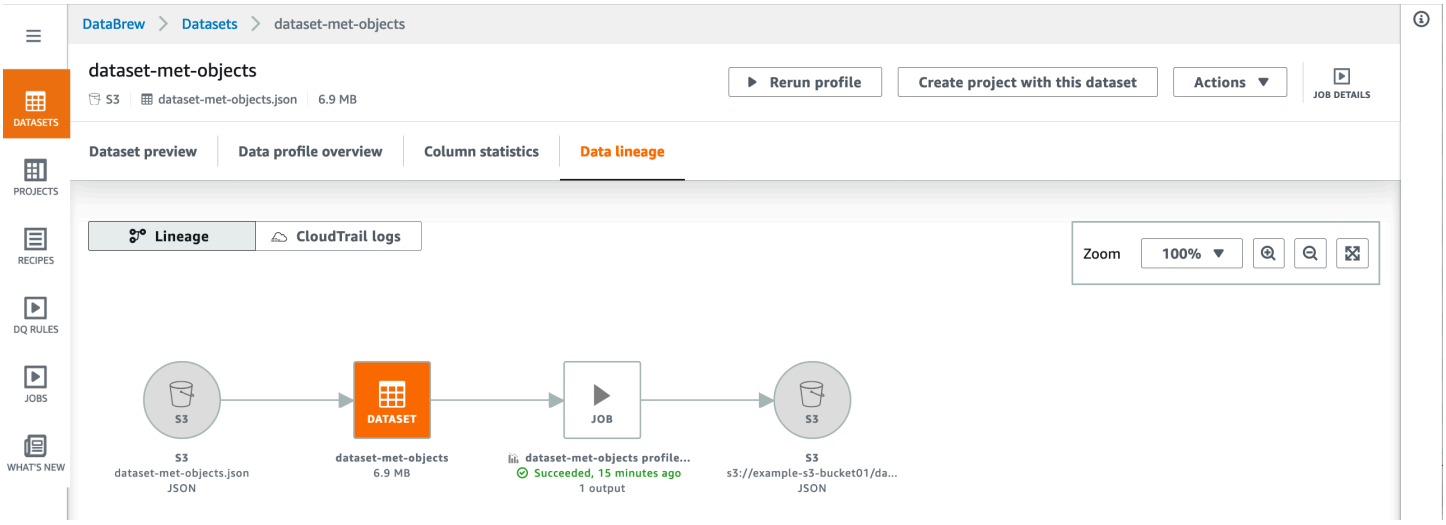


> **Note**
>
> Untuk membuat profil data, jalankan pekerjaan DataBrew profil di kumpulan data Anda. Untuk informasi tentang cara melakukan ini, lihat Langkah 5: Buat profil data.

Statistik kolom - Pada tab ini, Anda dapat menemukan statistik terperinci tentang setiap kolom dalam kumpulan data Anda, seperti yang ditunjukkan berikut.



Silsilah data - Tab ini menunjukkan representasi grafis tentang bagaimana dataset Anda dibuat dan bagaimana itu digunakan DataBrew, seperti yang ditunjukkan berikut.



Topik

- Menghapus dataset

Menghapus dataset

Jika Anda tidak lagi membutuhkan dataset, Anda dapat menghapusnya. Menghapus kumpulan data tidak memengaruhi sumber data yang mendasarinya dengan cara apa pun. Ini hanya menghapus informasi yang DataBrew digunakan untuk mengakses sumber data.

Anda tidak dapat menghapus kumpulan data jika DataBrew sumber daya lain bergantung padanya. Misalnya, jika saat ini Anda memiliki DataBrew proyek yang menggunakan kumpulan data, hapus proyek terlebih dahulu sebelum Anda menghapus kumpulan data.

Untuk menghapus kumpulan data, pilih Dataset dari panel navigasi. Pilih kumpulan data yang ingin Anda hapus, lalu untuk Tindakan, pilih Hapus.

Menghubungkan ke data Anda

Untuk informasi lebih lanjut tentang menghubungkan ke sumber data berikut, pilih bagian yang berlaku untuk Anda.

- AWS Glue Data Catalog— Anda dapat menggunakan Katalog Data untuk menentukan referensi ke objek data yang disimpan di AWS Cloud, termasuk layanan berikut:
  - Amazon Redshift
  - Aurora MySQL
  - Aurora PostgreSQL
  - Amazon RDS for MySQL
  - Amazon RDS for PostgreSQL

  DataBrew mengenali semua izin Lake Formation yang telah diterapkan ke sumber daya Katalog Data, sehingga DataBrew pengguna hanya dapat mengakses sumber daya ini jika mereka diizinkan.

  Untuk membuat kumpulan data, Anda menentukan nama database Katalog Data dan nama tabel. DataBrew mengurus detail koneksi lainnya.
- AWS Data Exchange — Anda dapat memilih dari ratusan sumber data pihak ketiga yang tersedia di AWS Data Exchange. Dengan berlangganan sumber data ini, Anda selalu memiliki versi data terbaru.

  Untuk membuat kumpulan data, Anda menentukan nama produk data Exchange Data yang Anda berlangganan atau berhak untuk digunakan.

- Koneksi driver JDBC — Anda dapat membuat kumpulan data dengan menghubungkan DataBrew ke sumber data. JDBC-compatible DataBrew mendukung koneksi ke sumber-sumber berikut melalui JDBC:
  - Amazon Redshift
  - Microsoft SQL Server
  - MySQL
  - Oracle
  - PostgreSQL
  - Kepingan salju

Topik

- Menggunakan driver dengan AWS Glue DataBrew
- Driver JDBC yang didukung

Menggunakan driver dengan AWS Glue DataBrew

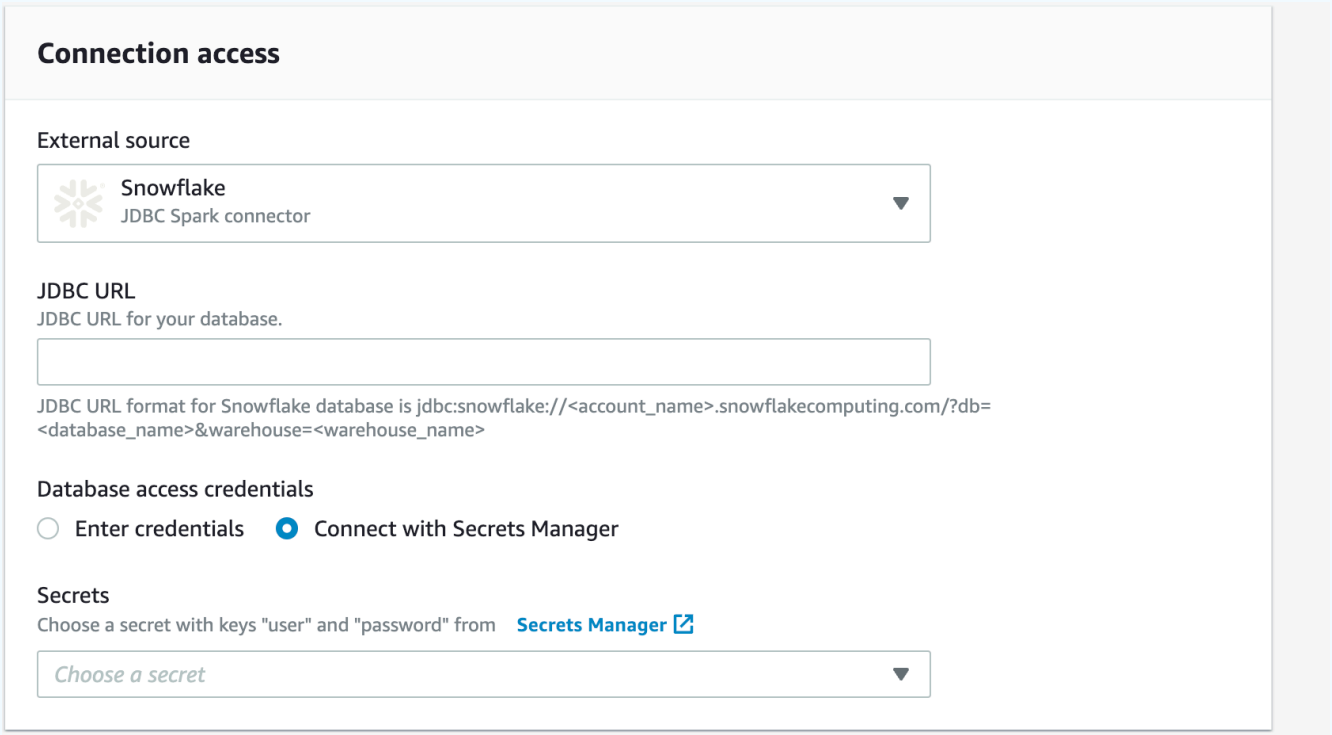
Driver database adalah file atau URL yang mengimplementasikan protokol koneksi database, misalnya Java Database Connectivity (JDBC). Driver berfungsi sebagai adaptor atau penerjemah antara sistem manajemen basis data tertentu (DBMS) dan sistem lain.

Dalam hal ini, memungkinkan AWS Glue DataBrew untuk terhubung ke data Anda. Kemudian Anda dapat mengakses objek database, seperti tabel atau tampilan, dari sumber data yang didukung. Sumber data yang Anda gunakan mungkin disebut database, gudang data, atau sesuatu yang lain. Namun, untuk tujuan dokumentasi ini kami menyebut semua penyedia data sebagai sumber data atau koneksi.

Untuk menggunakan driver JDBC atau file jar, unduh file atau file yang Anda butuhkan dan masukkan ke dalam ember S3. Peran IAM yang Anda gunakan untuk mengakses data harus memiliki izin baca untuk kedua file driver.

> **Note**
>
> With AWS Glue4.0, menghubungkan ke Snowflake sebagai sumber data didukung secara native. Anda tidak perlu menyediakan `jar` file khusus. Di AWS Glue DataBrew, pilih Snowflake sebagai koneksi sumber eksternal dan berikan URL instance

Snowflake Anda. URL akan menggunakan nama host dalam formulirhttps://account_identifier.snowflakecomputing.com.
Berikan kredensil akses data, nama database Snowflake, dan nama skema Snowflake. Selain itu, jika pengguna Snowflake Anda tidak memiliki set gudang default, Anda harus memberikan nama gudang.
Koneksi kepingan salju menggunakan AWS Secrets Manager rahasia untuk memberikan informasi kredensi. Proyek dan peran pekerjaan Anda harus memiliki izin untuk membaca rahasia ini.



Untuk menggunakan driver dengan DataBrew

1. Cari tahu versi sumber data yang Anda gunakan, menggunakan metode yang disediakan oleh produk.
2. Temukan versi terbaru dari konektor dan driver yang diperlukan. Anda dapat menemukan informasi ini di situs web penyedia data.
3. Unduh versi file JDBC yang diperlukan. Ini biasanya disimpan sebagai file Java Archives (.JAR).
4. Unggah driver dari konsol ke bucket S3 Anda atau berikan jalur S3 ke file.JAR Anda.
5. Masukkan detail koneksi dasar, misalnya kelas, contoh, dan sebagainya.

6. Masukkan informasi konfigurasi tambahan yang dibutuhkan sumber data Anda, misalnya informasi virtual private cloud (VPC).

Driver JDBC yang didukung

Produk	Versi yang didukung	Instruksi dan unduhan driver	Kueri SQL didukung
Microsoft SQL Server	v6.x atau lebih tinggi	Driver Microsoft JDBC untuk SQL Server	Tidak didukung
MySQL	v5.1 atau lebih tinggi	Konektor MySQL	Tidak didukung
Oracle	v11.2 atau lebih tinggi	Unduhan Oracle JDBC	Tidak didukung
PostgreSQL	v4.2.x atau lebih tinggi	Pengemudi PostgreSQL JDBC	Tidak didukung
Amazon Redshift	v4.1 atau lebih tinggi	Menghubungkan ke Amazon Redshift dengan JDBC	Didukung
	Untuk melihat versi	Untuk terhubung ke Snowflake, Anda memerlukan kedua hal berikut:	Didukung

Produk	Versi yang didukung	Instruksi dan unduhan driver	Kueri SQL didukung
Kepingan salju	Snowflake Anda, gunakan CURRENT ERSION seperti yang dijelaskan dalam dokumentasi Snowflake.	• Pengemudi Snowflake JDBC • Konektor Kepingan Salju untuk Spark	

Untuk terhubung ke database atau gudang data yang memerlukan versi driver yang berbeda dari apa yang didukung DataBrew secara native, Anda dapat menyediakan driver JDBC pilihan Anda. Driver harus kompatibel dengan JDK 8 atau Java 8. Untuk petunjuk tentang cara menemukan versi driver terbaru untuk database Anda, lihatMenggunakan driver dengan AWS Glue DataBrew.

Menghubungkan ke data dalam file teks dengan DataBrew

Anda dapat mengonfigurasi opsi format berikut untuk file input yang DataBrew mendukung:

- Comma-separated file nilai (CSV)
  - Pembatas

    Pembatas default adalah koma untuk file.csv. Jika file Anda menggunakan pembatas yang berbeda, pilih pembatas untuk pembatas CSV di bagian Konfigurasi tambahan saat Anda membuat kumpulan data. Pembatas berikut didukung untuk file.csv:

    - Koma (,)
    - Usus besar (:)
    - Semi-colon (;)

- Pipa (|)
- Tab (\t)
- Karet (^)
- Backslash (\)
- Spasi

- Nilai header kolom

  File CSV Anda dapat menyertakan baris header sebagai baris pertama file. Jika tidak, DataBrew buat baris header untuk Anda.

  - Jika file CSV Anda menyertakan baris header, pilih Perlakukan baris pertama sebagai header. Jika Anda melakukannya, baris pertama file CSV Anda diperlakukan sebagai berisi nilai header kolom.
  - Jika file CSV Anda tidak menyertakan baris header, pilih Tambahkan header default. Jika Anda melakukannya, DataBrew buat baris header untuk file dan tidak memperlakukan baris data pertama Anda sebagai berisi nilai header. Header yang DataBrew dibuat terdiri dari garis bawah dan angka untuk setiap kolom dalam file, dalam format`Column_1`,, `Column_2Column_3`, dan sebagainya.

- Berkas JSON

  DataBrew mendukung dua format untuk file JSON, JSON Lines dan dokumen JSON. File JSON Lines berisi satu baris per baris. Dalam file dokumen JSON, semua baris terkandung dalam struktur JSON tunggal atau array. Anda dapat menentukan jenis file JSON Anda di bagian Konfigurasi tambahan saat Anda membuat kumpulan data JSON. Format defaultnya adalah JSON Lines.

- File Excel

  Berikut ini berlaku untuk lembar Excel di DataBrew:

  - Pemuatan lembar Excel

    Secara default, DataBrew memuat lembar pertama di file Excel Anda. Namun, Anda dapat menentukan nomor lembar atau nama lembar yang berbeda di bagian Konfigurasi tambahan saat Anda membuat kumpulan data Excel.

  - Nilai header kolom

    Lembar Excel Anda dapat menyertakan baris header sebagai baris pertama file, tetapi jika tidak, DataBrew akan membuat baris header untuk Anda.

- Jika lembar Excel Anda menyertakan baris header, pilih Perlakukan baris pertama sebagai header. Jika Anda melakukannya, baris pertama lembar Excel Anda diperlakukan sebagai berisi nilai header kolom.
- Jika file Excel Anda tidak menyertakan baris header, pilih Tambahkan header default. Dengan melakukan ini, Anda menentukan yang DataBrew harus membuat baris header untuk file dan tidak memperlakukan baris pertama data Anda sebagai berisi nilai header. Header yang DataBrew dibuat terdiri dari garis bawah dan angka untuk setiap kolom dalam file, dalam format`Column_1`,, `Column_2``Column_3`, dan sebagainya.

Menghubungkan data dalam beberapa file di Amazon S3

Dengan DataBrew konsol, Anda dapat menavigasi bucket dan folder Amazon S3 dan memilih file untuk kumpulan data Anda. Namun, kumpulan data tidak perlu dibatasi pada satu file.

Misalkan Anda memiliki bucket S3 bernama `my-databrew-bucket` yang berisi folder bernama`databrew-input`. Dalam folder itu, misalkan Anda memiliki sejumlah file JSON, semua dengan format file dan ekstensi `.json` file yang sama. Di konsol, Anda dapat menentukan URL sumber dari`s3://my-databrew-bucket/databrew-input/`. Di DataBrew konsol, Anda kemudian dapat memilih folder ini. Dataset Anda terdiri dari semua file JSON di folder itu.

DataBrew dapat memproses semua file dalam folder S3, tetapi hanya jika kondisi berikut benar:

- Semua file dalam folder memiliki format yang sama.
- Semua file dalam folder memiliki ekstensi file yang sama.

Untuk informasi selengkapnya tentang format dan ekstensi file yang didukung, lihatDataBrew input formats.

Skema saat menggunakan banyak file sebagai kumpulan data

Saat menggunakan banyak file sebagai DataBrew kumpulan data, skema harus sama di semua file. Jika tidak, Project Workspace secara otomatis mencoba memilih salah satu skema dari beberapa file dan mencoba menyesuaikan sisa file kumpulan data dengan skema itu. Perilaku ini menghasilkan tampilan yang ditampilkan selama Project Workspace menjadi tidak teratur, dan akibatnya, output pekerjaan juga akan tidak teratur.

Jika file Anda harus memiliki skema yang berbeda, Anda perlu membuat beberapa kumpulan data dan memprofilkannya secara terpisah.

Menggunakan jalur berparameter untuk Amazon S3

Dalam beberapa kasus, Anda mungkin ingin membuat kumpulan data dengan file yang mengikuti konvensi penamaan tertentu, atau kumpulan data yang dapat menjangkau beberapa folder Amazon S3. Atau Anda mungkin ingin menggunakan kembali kumpulan data yang sama untuk data terstruktur identik yang dihasilkan secara berkala di lokasi S3 dengan jalur yang bergantung pada parameter tertentu. Contohnya adalah jalur yang diberi nama untuk tanggal produksi data.

DataBrew mendukung pendekatan ini dengan jalur S3 berparameter. Jalur berparameter adalah URL Amazon S3 yang berisi ekspresi reguler atau parameter jalur khusus, atau keduanya.

Mendefinisikan kumpulan data dengan jalur S3 menggunakan ekspresi reguler

Ekspresi reguler di jalur dapat berguna untuk mencocokkan beberapa file dari satu atau beberapa folder dan pada saat yang sama menyaring file yang tidak terkait di folder tersebut.

Berikut adalah beberapa contoh:

- Tentukan kumpulan data termasuk semua file JSON dari folder yang namanya dimulai dengan. `invoice`
- Tentukan kumpulan data termasuk semua file dalam folder dengan `2020` namanya.

Anda dapat menerapkan jenis pendekatan ini dengan menggunakan ekspresi reguler di jalur dataset S3. Ekspresi reguler ini dapat menggantikan substring apa pun di kunci URL S3 (tetapi bukan nama bucket).

Sebagai contoh kunci dalam URL S3, lihat yang berikut ini. Di sini, `my-bucket` adalah nama ember, US East (Ohio) adalah AWS Region, dan `puppy.png` merupakan nama kuncinya.

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

Dalam jalur S3 berparameter, karakter apa pun di antara dua tanda kurung sudut (<dan>) diperlakukan sebagai ekspresi reguler. Dua contoh adalah sebagai berikut:

- `s3://my-databrew-bucket/databrew-input/invoice<.*>/data.json`cocok dengan semua file bernama`data.json`, dalam semua subfolder `databrew-input` yang namanya dimulai dengan`invoice`.

- `s3://my-databrew-bucket/databrew-input/<.*>2020<.*>/`mencocokkan semua file `2020` dalam folder dengan namanya.

Dalam contoh ini, `.*` cocok dengan nol atau lebih karakter.

> **Note**
>
> Anda hanya dapat menggunakan ekspresi reguler di bagian kunci jalur S3—bagian yang mengikuti nama bucket. Jadi, `s3://my-databrew-bucket/<.*>-input/` valid, tetapi `s3://my-<.*>-bucket/<.*>-input/` tidak.

Kami menyarankan Anda menguji ekspresi reguler Anda untuk memastikan bahwa mereka hanya cocok dengan URL S3 yang Anda inginkan, dan bukan yang tidak Anda inginkan.

Berikut adalah beberapa contoh ekspresi reguler lainnya:

- `<\d{2}>`cocok dengan string yang terdiri dari tepat dua digit berturut-turut, misalnya `07` atau`03`, tetapi tidak. `1a2`
- `<[a-z]+.*>`cocok dengan string yang dimulai dengan satu atau lebih huruf Latin kecil dan memiliki nol atau lebih karakter lain setelahnya. Contohnya adalaha`3`,`abc/def`, ataua`-z`, tetapi tidak`A2`.
- `<[^/]+>`cocok dengan string yang berisi karakter apa pun kecuali garis miring (/). Dalam URL S3, garis miring digunakan untuk memisahkan folder di jalur.
- `<.*=.*>`cocok dengan string yang berisi tanda sama dengan (=), misalnya,, atau `month=02` `abc/day=2=10`, tetapi tidak. `test`
- `<\d.*\d>`cocok dengan string yang dimulai dan diakhiri dengan digit dan dapat memiliki karakter lain di antara digit, misalnya,`1abc2`, atau `01-02-032020/Jul/21`, tetapi tidak`123a`.

Mendefinisikan kumpulan data dengan jalur S3 menggunakan parameter khusus

Mendefinisikan kumpulan data berparameter menggunakan parameter khusus menawarkan keuntungan dibandingkan menggunakan ekspresi reguler saat Anda mungkin ingin memberikan parameter untuk lokasi S3:

- Anda dapat mencapai hasil yang sama dengan ekspresi reguler, tanpa perlu mengetahui sintaks untuk ekspresi reguler. Anda dapat menentukan parameter menggunakan istilah yang sudah dikenal seperti “dimulai dengan” dan “berisi.”
- Saat Anda menentukan kumpulan data dinamis menggunakan parameter di jalur, Anda dapat menyertakan rentang waktu dalam definisi Anda, seperti “bulan lalu” atau “24 jam terakhir.” Dengan begitu, definisi dataset Anda akan digunakan nanti dengan data baru yang masuk.

Berikut adalah beberapa contoh kapan Anda mungkin ingin menggunakan kumpulan data dinamis:

- Untuk menghubungkan beberapa file yang dipartisi berdasarkan tanggal terakhir diperbarui atau atribut bermakna lainnya ke dalam satu kumpulan data. Anda kemudian dapat menangkap atribut partisi ini sebagai kolom tambahan dalam kumpulan data.
- Untuk membatasi file dalam kumpulan data ke lokasi S3 yang memenuhi kondisi tertentu. Misalnya, misalkan jalur S3 Anda berisi folder berbasis tanggal seperti. `folder/2021/04/01/` Dalam hal ini, Anda dapat membuat parameter tanggal dan membatasinya ke kisaran tertentu seperti “antara 01 Mar 2021 dan 01 Apr 2021" atau “Minggu lalu.”

Untuk menentukan jalur menggunakan parameter, tentukan parameter dan tambahkan ke jalur Anda menggunakan format berikut:

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

> **Note**
>
> Seperti halnya ekspresi reguler di jalur S3, Anda hanya dapat menggunakan parameter di bagian kunci jalur—bagian yang mengikuti nama bucket.

Dua bidang diperlukan dalam definisi parameter, nama dan jenis. Tipenya bisa berupa String, Number, atau Date. Parameter tipe Tanggal harus memiliki definisi format tanggal sehingga DataBrew dapat menafsirkan dan membandingkan nilai tanggal dengan benar. Secara opsional, Anda dapat menentukan kondisi pencocokan untuk parameter. Anda juga dapat memilih untuk menambahkan nilai parameter yang cocok sebagai kolom ke kumpulan data Anda saat sedang dimuat oleh DataBrew pekerjaan atau sesi interaktif.

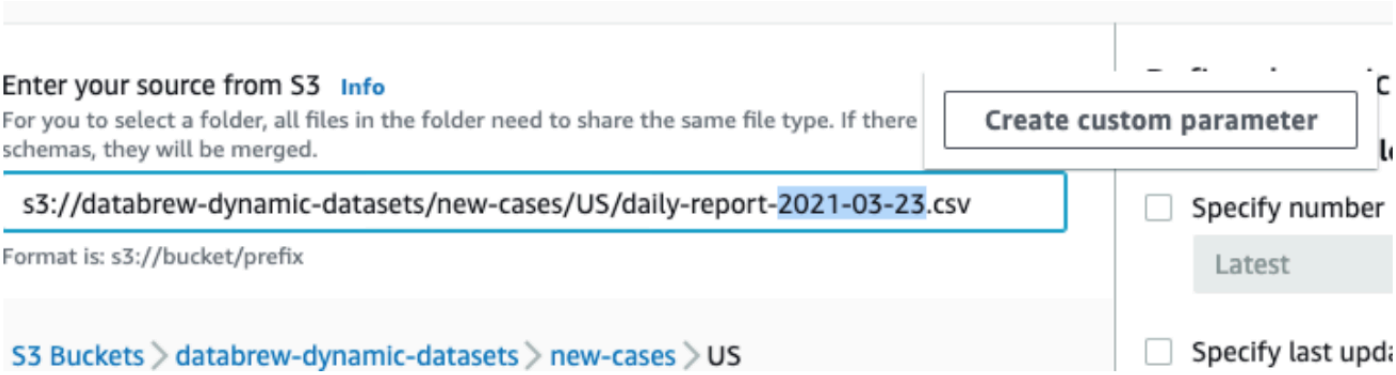
Contoh

Mari kita pertimbangkan contoh mendefinisikan dataset dinamis menggunakan parameter di konsol. DataBrew Dalam contoh ini, asumsikan bahwa data input secara teratur ditulis ke dalam bucket S3 menggunakan lokasi seperti ini:

- `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`

Ada dua bagian dinamis di sini: kode negara, seperti AS, dan tanggal dalam nama file seperti 2021-03-30. Di sini, Anda dapat menerapkan resep pembersihan yang sama untuk semua file. Katakanlah Anda ingin melakukan pekerjaan pembersihan Anda setiap hari. Berikut ini adalah bagaimana Anda dapat menentukan jalur berparameter untuk skenario ini:

1. Arahkan ke file tertentu.
2. Kemudian pilih bagian yang bervariasi, seperti tanggal, dan ganti dengan parameter. Dalam hal ini, ganti tanggal.

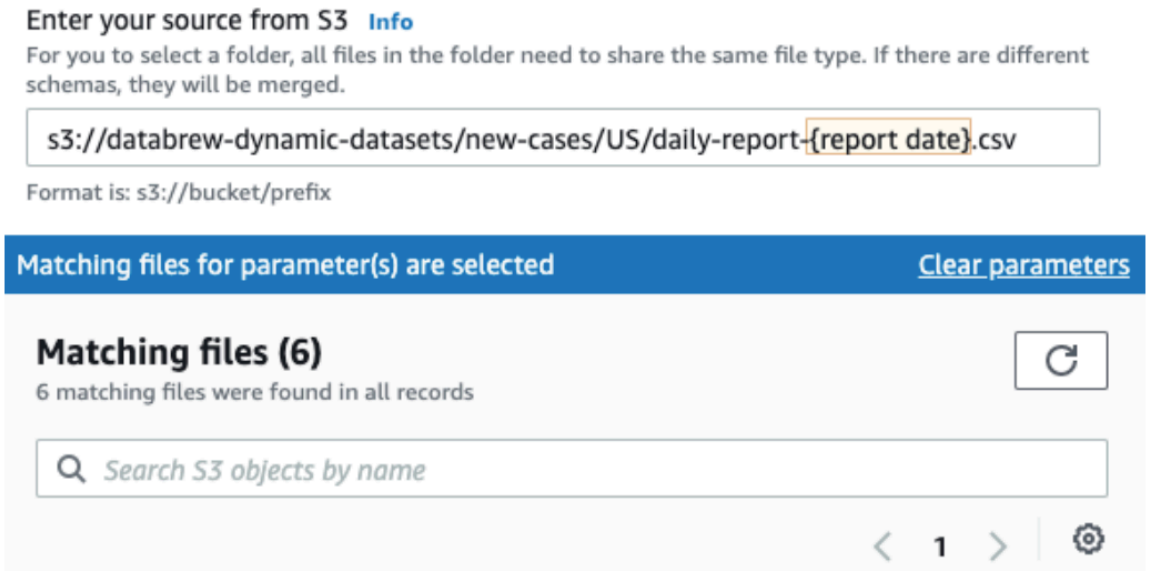


3. Buka menu konteks (klik kanan) untuk Buat parameter khusus dan atur properti untuknya:
   - Nama: tanggal laporan
   - Tipe: Tanggal
   - Format tanggal: yyyy- MM-dd (dipilih dari format yang telah ditentukan)
   - Kondisi (Rentang waktu): 24 jam terakhir
   - Tambahkan sebagai kolom: true (dicentang)

   Simpan bidang lain pada nilai defaultnya.

4. Pilih Buat.

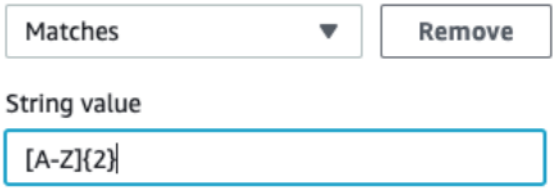
Setelah Anda melakukannya, Anda melihat jalur yang diperbarui, seperti pada tangkapan layar berikut.



Sekarang Anda dapat melakukan hal yang sama untuk kode negara dan membuat parameter sebagai berikut:

- Nama: kode negara
- Tipe: String
- Tambahkan sebagai kolom: true (dicentang)

Anda tidak perlu menentukan kondisi jika semua nilai relevan. Di `new-cases` folder, misalnya, kami hanya memiliki subfolder dengan kode negara, jadi tidak perlu kondisi. Jika Anda memiliki folder lain untuk dikecualikan, Anda dapat menggunakan kondisi berikut.



Pendekatan ini membatasi subfolder kasus baru untuk berisi dua huruf Latin kapital.

Setelah parameterisasi ini, Anda hanya memiliki file yang cocok di kumpulan data kami dan dapat memilih Buat Dataset.

> **Note**
>
> Saat Anda menggunakan rentang waktu relatif dalam kondisi, rentang waktu dievaluasi saat kumpulan data dimuat. Ini benar apakah rentang waktu yang telah ditentukan seperti “24 jam terakhir” atau rentang waktu khusus seperti “5 hari yang lalu”. Pendekatan evaluasi ini berlaku apakah kumpulan data dimuat selama inisialisasi sesi interaktif atau selama awal pekerjaan.

Setelah Anda memilih Buat Dataset, dataset dinamis Anda siap digunakan. Sebagai contoh, Anda dapat menggunakannya terlebih dahulu untuk membuat proyek dan menentukan resep pembersihan menggunakan DataBrew sesi interaktif. Kemudian Anda dapat membuat pekerjaan yang dijadwalkan untuk berjalan setiap hari. Pekerjaan ini mungkin menerapkan resep pembersihan ke file kumpulan data yang memenuhi kondisi parameter Anda pada saat pekerjaan dimulai.

Kondisi yang didukung untuk kumpulan data dinamis

Anda dapat menggunakan kondisi untuk memfilter file S3 yang cocok menggunakan parameter atau atribut tanggal modifikasi terakhir.

Berikut ini, Anda dapat menemukan daftar kondisi yang didukung untuk setiap jenis parameter.

Kondisi yang digunakan dengan parameter String

Nama dalam DataBrew SDK	Sinonim SDK	Nama di DataBrew konsol	Deskripsi
adalah	persamaan, ==	Tepat	Nilai parameter sama dengan nilai yang diberikan dalam kondisi.
tidak	bukan persamaan,! =	Is not	Nilai parameter tidak sama dengan nilai yang diberikan dalam kondisi.
mengandung		Contains	Nilai string parameter berisi nilai yang

Nama dalam DataBrew SDK	Sinonim SDK	Nama di DataBrew konsol	Deskripsi
			disediakan dalam kondisi.
tidak mengandung		Tidak mengandung	Nilai string parameter tidak berisi nilai yang disediakan dalam kondisi.
mulai_dengan		Starts with	Nilai string parameter dimulai dengan nilai yang disediakan dalam kondisi.
bukan starts_with		Tidak dimulai dengan	Nilai string parameter tidak dimulai dengan nilai yang disediakan dalam kondisi.
berakhir_dengan		Ends with	Nilai string parameter berakhir dengan nilai yang disediakan dalam kondisi.
tidak berakhir_dengan		Tidak berakhir dengan	Nilai string parameter tidak berakhir dengan nilai yang disediakan dalam kondisi.
korek api		Cocok	Nilai parameter cocok dengan ekspresi reguler yang disediakan dalam kondisi.

Nama dalam DataBrew SDK	Sinonim SDK	Nama di DataBrew konsol	Deskripsi
tidak cocok		Tidak cocok	Nilai parameter tidak cocok dengan ekspresi reguler yang disediakan dalam kondisi.

> **Note**
>
> Semua kondisi untuk parameter String menggunakan perbandingan peka huruf besar/kecil. Jika Anda tidak yakin tentang kasus yang digunakan di jalur S3, Anda dapat menggunakan kondisi “cocok” dengan nilai ekspresi reguler yang dimulai dengan`(?i)`. Melakukan hal ini menghasilkan perbandingan case-insensitive.
> Misalnya, misalkan Anda ingin parameter string Anda dimulai`abc`, tetapi `Abc` atau `ABC` juga dimungkinkan. Dalam hal ini, Anda dapat menggunakan kondisi “cocok” dengan `(?i)^abc` sebagai nilai kondisi.

Kondisi yang digunakan dengan parameter Angka

Nama dalam DataBrew SDK	Sinonim SDK	Nama di DataBrew konsol	Deskripsi
adalah	persamaan, ==	Tepat	Nilai parameter sama dengan nilai yang diberikan dalam kondisi.
tidak	bukan persamaan,! =	Is not	Nilai parameter tidak sama dengan nilai yang diberikan dalam kondisi.
kurang_dari	lt, <	Kurang dari	Nilai numerik parameter kurang dari

Nama dalam DataBrew SDK	Sinonim SDK	Nama di DataBrew konsol	Deskripsi
			nilai yang disediakan dalam kondisi.
less_than_equal	lte, <=	Kurang dari atau sama dengan	Nilai numerik parameter kurang dari atau sama dengan nilai yang diberikan dalam kondisi.
lebih besar_dari	gt, >	Lebih besar dari	Nilai numerik parameter lebih besar dari nilai yang diberikan dalam kondisi.
lebih besar_tha n_equal	gte, >=	Lebih besar dari atau sama dengan	Nilai numerik parameter lebih besar dari atau sama dengan nilai yang diberikan dalam kondisi.

Kondisi yang digunakan dengan parameter Tanggal

Nama dalam DataBrew SDK	Nama di DataBrew konsol	Format nilai kondisi (SDK)	Deskripsi
setelah	Mulai	Format tanggal ISO 8601 seperti atau `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	Nilai parameter tanggal adalah setelah tanggal yang diberikan dalam kondisi.

Nama dalam DataBrew SDK	Nama di DataBrew konsol	Format nilai kondisi (SDK)	Deskripsi
sebelumnya	Akhiri	Format tanggal ISO 8601 seperti atau `2021-03-30T01:00:00Z` `2021-03-30T01:00-07:00`	Nilai parameter tanggal sebelum tanggal yang diberikan dalam kondisi.
relatif_after	Mulai (relatif)	Jumlah satuan waktu positif atau negatif, seperti `-48h` atau`+7d`.	Nilai parameter tanggal adalah setelah tanggal relatif yang diberikan dalam kondisi. Tanggal relatif dievaluasi saat dataset dimuat, baik saat sesi interaktif diinisialisasi atau saat pekerjaan terkait dimulai. Ini adalah momen yang disebut “sekarang” dalam contoh.

Nama dalam DataBrew SDK	Nama di DataBrew konsol	Format nilai kondisi (SDK)	Deskripsi
relatif_before	Akhir (relatif)	Jumlah satuan waktu positif atau negatif, seperti `-48h` atau`+7d`.	Nilai parameter tanggal sebelum tanggal relatif yang disediakan dalam kondisi. Tanggal relatif dievaluasi saat dataset dimuat, baik saat sesi interakti f diinisialisasi atau saat pekerjaan terkait dimulai. Ini adalah momen yang disebut “sekarang” dalam contoh.

Jika Anda menggunakan SDK, berikan tanggal relatif dalam format berikut:`±{number_of_time_units}{time_unit}`. Anda dapat menggunakan unit waktu ini:

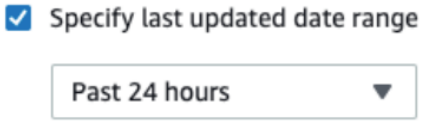
- -1h (1 jam lalu)
- +2d (2 hari dari sekarang)
- -120m (120 menit lalu)
- 5000s (5.000 detik dari sekarang)
- -3w (3 minggu lalu)
- +4M (4 bulan dari sekarang)
- -1y (1 tahun lalu)

Tanggal relatif dievaluasi saat dataset dimuat, baik saat sesi interaktif diinisialisasi atau saat pekerjaan terkait dimulai. Ini adalah momen yang disebut “sekarang” dalam contoh sebelumnya.

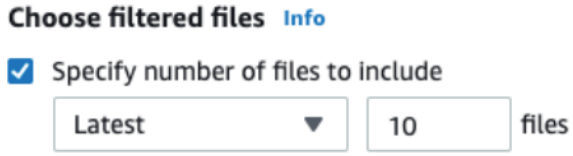
Mengkonfigurasi pengaturan untuk kumpulan data dinamis

Selain menyediakan jalur S3 berparameter, Anda dapat mengonfigurasi pengaturan lain untuk kumpulan data dengan banyak file. Pengaturan ini memfilter file S3 berdasarkan tanggal modifikasi terakhir dan membatasi jumlah file.

Mirip dengan menyetel parameter tanggal di jalur, Anda dapat menentukan rentang waktu saat file yang cocok diperbarui dan hanya menyertakan file tersebut ke dalam kumpulan data Anda. Anda dapat menentukan rentang ini menggunakan tanggal absolut seperti “30 Maret 2021" atau rentang relatif seperti “24 jam terakhir”.



Untuk membatasi jumlah file yang cocok, pilih sejumlah file yang lebih besar dari 0 dan apakah Anda menginginkan file pencocokan terbaru atau tertua.



Jenis Data

Data untuk setiap kolom kumpulan data Anda dikonversi ke salah satu tipe data berikut:

- byte - nomor integer bertanda 1-byte. Kisaran angka adalah dari -128 hingga 127.
- pendek — nomor integer bertanda 2-byte. Kisaran angka adalah dari -32768 hingga 32767.
- bilangan bulat — nomor integer bertanda 4-byte. Kisaran angka adalah dari -2147483648 hingga 2147483647.
- panjang — nomor bilangan bulat bertanda 8-byte. Kisaran angka adalah dari -9223372036854775808 hingga 9223372036854775807.
- float — nomor floating point presisi tunggal 4-byte.
- ganda — nomor titik mengambang presisi ganda 8-byte.
- desimal — Menandatangani angka desimal dengan total 38 digit dan 18 digit setelah titik desimal.
- string - Nilai string karakter.

- boolean — Tipe Boolean memiliki salah satu dari dua nilai yang mungkin: `true` dan `false` atau `yes` dan `no`.
- timestamp — Nilai yang terdiri dari bidang tahun, bulan, hari, jam, menit, dan detik.
- tanggal — Nilai yang terdiri dari bidang tahun, bulan dan hari.

Tipe data tingkat lanjut

Tipe data lanjutan adalah tipe data yang DataBrew mendeteksi dalam kolom string dalam proyek, dan oleh karena itu bukan bagian dari kumpulan data. Untuk informasi tentang tipe data lanjutan, lihat Tipe data lanjutan.

Tipe data tingkat lanjut

Tipe data lanjutan adalah tipe data yang DataBrew mendeteksi dalam kolom string dalam proyek dengan cara pencocokan pola. Saat Anda mengklik kolom string, kolom ditandai sebagai tipe data lanjutan yang sesuai jika 50% atau lebih nilai di kolom memenuhi kriteria untuk tipe data tersebut.

Tipe data yang DataBrew dapat dideteksi adalah:

- Date/timestamp
- SSN
- Nomor telepon
- Email
- Kartu kredit
- Gender
- Alamat IP
- URL
- Kode pos
- Negara
- Mata Uang
- Status
- Kota

Anda dapat menggunakan transformasi berikut untuk bekerja dengan tipe data lanjutan:

- GET_ADVANCED_DATATYPE: Diberikan kolom string, mengidentifikasi tipe data lanjutan dari kolom, jika ada.
- EXTRACT_ADVANCED_DATATYPE_DETAILS: Mengekstrak detail untuk tipe data lanjutan.
- ADVANCED_DATATYPE_FILTER: Memfilter kolom sumber saat ini berdasarkan deteksi tipe data lanjutan.
- ADVANCED_DATATYPE_FLAG: Membuat kolom bendera baru berdasarkan nilai untuk kolom sumber saat ini.

Memvalidasi kualitas data di AWS Glue DataBrew

Untuk memastikan kualitas kumpulan data Anda, Anda dapat menentukan daftar aturan kualitas data dalam kumpulan aturan. Kumpulan aturan adalah seperangkat aturan yang membandingkan metrik data yang berbeda dengan nilai yang diharapkan. Jika salah satu kriteria aturan tidak terpenuhi, kumpulan aturan secara keseluruhan gagal validasi. Anda kemudian dapat memeriksa hasil individu untuk setiap aturan. Untuk aturan apa pun yang menyebabkan kegagalan validasi, Anda dapat melakukan koreksi yang diperlukan dan memvalidasi ulang.

Contoh aturan meliputi:

- Nilai dalam kolom `"APY"` adalah antara 0 dan 100
- Jumlah nilai yang hilang di kolom `group_name` tidak melebihi 5%

Anda dapat menentukan setiap aturan untuk kolom individual atau menerapkannya secara independen ke beberapa kolom yang dipilih, misalnya:

- Nilai maks tidak melebihi 100 untuk kolom`"rate"`,`"pay"`,`"increase"`.

Aturan dapat terdiri dari beberapa pemeriksaan sederhana. Anda dapat menentukan apakah semuanya harus benar atau apa pun, misalnya:

- Nilai dalam kolom `"ProductId"` harus dimulai dengan `"asin-"` DAN panjang nilai di kolom `"ProductId"` adalah 32.

Anda dapat memverifikasi aturan terhadap nilai agregat seperti`max`,`min`, atau `number of duplicate values` di mana hanya ada satu nilai yang dibandingkan, atau nilai nonagregat di setiap baris kolom. Dalam kasus terakhir, Anda juga dapat menentukan ambang batas “lewat” seperti`value in columnA > value in columnB for at least 95% of rows`.

Seperti informasi profil, Anda dapat menentukan aturan kualitas data tingkat kolom hanya untuk kolom tipe sederhana, seperti string dan angka. Anda tidak dapat menentukan aturan kualitas data untuk kolom tipe kompleks, seperti array atau struktur. Untuk detail selengkapnya tentang bekerja dengan informasi profil, lihat[Membuat dan bekerja dengan AWS Glue DataBrew lowongan kerja profil](#).

Memvalidasi aturan kualitas data

Setelah kumpulan aturan ditentukan, Anda dapat menambahkannya ke pekerjaan profil untuk validasi. Anda dapat menentukan lebih dari satu kumpulan aturan untuk kumpulan data.

Misalnya, satu set aturan mungkin berisi aturan dengan kriteria minimal yang dapat diterima. Kegagalan validasi untuk kumpulan aturan itu mungkin berarti bahwa data tidak dapat diterima untuk digunakan lebih lanjut. Contohnya adalah nilai yang hilang di kolom kunci dari kumpulan data yang digunakan untuk pelatihan pembelajaran mesin. Anda dapat menggunakan kumpulan aturan kedua dengan aturan yang lebih ketat untuk memverifikasi apakah kumpulan data memiliki kualitas yang baik sehingga tidak diperlukan pembersihan.

Anda dapat menerapkan satu atau beberapa kumpulan aturan yang ditentukan untuk kumpulan data tertentu dalam konfigurasi pekerjaan profil. Ketika pekerjaan profil berjalan, itu menghasilkan laporan validasi selain profil data. Laporan validasi tersedia di lokasi yang sama dengan data profil Anda. Seperti informasi profil, Anda dapat menjelajahi hasilnya di DataBrew konsol. Dalam tampilan Detail kumpulan data, pilih tab Kualitas Data untuk melihat hasilnya. Untuk detail selengkapnya tentang bekerja dengan informasi profil, lihatMembuat dan bekerja dengan AWS Glue DataBrew lowongan kerja profil.

Bertindak atas hasil validasi

Saat pekerjaan DataBrew profil selesai, DataBrew kirimkan CloudWatch acara Amazon dengan detail pekerjaan yang dijalankan. Jika Anda juga mengonfigurasi pekerjaan Anda untuk memvalidasi aturan kualitas data, DataBrew mengirimkan peristiwa untuk setiap kumpulan aturan yang divalidasi. Acara ini berisi hasilnya (`SUCCEEDED`,`FAILED`, atau`ERROR`) dan tautan ke laporan validasi kualitas data terperinci. Anda kemudian dapat mengotomatiskan tindakan lebih lanjut dengan menjalankan tindakan berikutnya tergantung pada status validasi. Untuk informasi selengkapnya tentang menghubungkan peristiwa ke tindakan target, seperti notifikasi Amazon SNS, pemanggilan AWS Lambda fungsi, dan lainnya, lihat Memulai Amazon. EventBridge

Berikut ini adalah contoh peristiwa Hasil DataBrew Validasi:

```
{
  "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"
  }
}
```

Anda dapat menggunakan atribut peristiwa sepertidetail-type, source dan properti bersarang dari detail atribut untuk membuat pola peristiwa di Amazon Eventbridge. Misalnya pola acara untuk mencocokkan semua validasi yang gagal dari DataBrew pekerjaan apa pun akan terlihat seperti ini:

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

Untuk contoh membuat kumpulan aturan dan memvalidasi aturannya, lihat. Membuat ruleset dengan aturan kualitas data Untuk informasi selengkapnya tentang bekerja dengan CloudWatch acara di DataBrew, lihat Mengotomatisasi DataBrew dengan Acara CloudWatch

Membuat ruleset dengan aturan kualitas data

Dalam prosedur berikut, Anda dapat menemukan contoh membuat kumpulan aturan dan menerapkannya ke kumpulan data. Kumpulan aturan adalah seperangkat aturan yang membandingkan metrik data yang berbeda dengan nilai yang diharapkan. Anda kemudian dapat menggunakan kumpulan aturan ini dalam pekerjaan profil untuk memvalidasi aturan kualitas data yang disertakan.

Untuk membuat contoh ruleset dengan aturan kualitas data

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol di https://console.aws.amazon.com/databrew/.
2. Pilih ATURAN DQ dari panel navigasi, lalu pilih Buat kumpulan aturan kualitas data.
3. Masukkan nama untuk kumpulan aturan Anda. Secara opsional, masukkan deskripsi untuk kumpulan aturan Anda.
4. Di bawah Kumpulan data terkait, pilih kumpulan data untuk dikaitkan dengan kumpulan aturan.

   Setelah memilih kumpulan data, Anda dapat melihat panel pratinjau Dataset di sebelah kanan.
5. Gunakan pratinjau di panel pratinjau Dataset untuk menjelajahi nilai dan skema untuk kumpulan data saat Anda menentukan aturan kualitas data yang akan dibuat. Pratinjau dapat memberi Anda wawasan tentang potensi masalah yang mungkin Anda miliki dengan data.

   Beberapa sumber data, seperti database, tidak mendukung pratinjau data. Dalam hal ini, Anda dapat menjalankan pekerjaan profil tanpa memvalidasi aturan kualitas data terlebih dahulu. Kemudian Anda bisa mendapatkan informasi tentang skema data dan distribusi nilai dengan menggunakan profil data.
6. Periksa tab Rekomendasi, yang mencantumkan beberapa saran aturan yang dapat Anda gunakan saat membuat kumpulan aturan Anda. Anda dapat memilih semua, beberapa, atau tidak ada rekomendasi.

   Setelah memilih rekomendasi yang relevan, pilih Tambahkan ke set aturan.

   Ini akan menambah aturan ke set aturan Anda. Periksa dan modifikasi parameter jika diperlukan. Perhatikan bahwa hanya kolom tipe sederhana seperti string, angka, dan boolean yang dapat digunakan dalam aturan kualitas data.
7. Pilih Tambahkan aturan lain untuk menambahkan aturan yang tidak tercakup oleh rekomendasi. Anda dapat mengubah nama aturan agar lebih mudah menafsirkan hasil validasi nanti.
8. Gunakan cakupan pemeriksaan kualitas data untuk memilih apakah kolom individu akan dipilih per setiap pemeriksaan dalam aturan ini atau apakah mereka harus diterapkan ke sekelompok kolom yang Anda pilih. Misalnya, jika kumpulan data Anda memiliki beberapa kolom numerik yang seharusnya memiliki nilai antara 0 dan 100, Anda dapat menentukan aturan satu kali dan memilih semua kolom ini untuk diperiksa oleh aturan ini.
9. Jika aturan Anda akan memiliki lebih dari satu pemeriksaan, maka dalam dropdown kriteria keberhasilan Aturan, pilih apakah semua cek harus dipenuhi atau mana yang memenuhi kriteria.

10. Pilih pemeriksaan yang akan dilakukan untuk memverifikasi aturan ini di dropdown pemeriksaan kualitas data. Untuk informasi selengkapnya tentang pemeriksaan yang tersedia, lihatCek yang tersedia.
11. Jika Anda memilih Individual cek untuk setiap kolom dalam lingkup pemeriksaan kualitas data, pilih kolom. Pilih atau ketik nama kolom untuk pemeriksaan ini.
12. Pilih parameter tergantung pada cek. Beberapa kondisi hanya menerima nilai kustom yang disediakan dan beberapa juga mendukung referensi ke kolom lain.
13. Jika Anda memilih cek untuk nilai Kolom seperti Mengandung kondisi untuk nilai string, maka Anda dapat menentukan ambang batas “lulus”. Misalnya, jika Anda ingin setidaknya 95 persen nilai memenuhi kondisi, Anda harus memilih Lebih Besar dari sama dengan Kondisi ambang batas, masukkan 95 sebagai Ambang dan biarkan “% (persen) baris” di dropdown berikutnya di bagian Threshold. Atau jika Anda ingin tidak lebih dari 10 baris di mana nilai hilang kondisi benar, maka Anda dapat memilih Kurang dari sama dengan sebagai Kondisi, masukkan 10 untuk Threshold dan pilih baris di dropdown berikutnya. Harap dicatat bahwa Anda mungkin mendapatkan hasil yang berbeda jika Anda menggunakan sampel dengan ukuran berbeda selama validasi.
14. Tambahkan lebih banyak aturan jika diperlukan.
15. Pilih Buat set aturan.

Membuat pekerjaan profil menggunakan ruleset

Setelah membuat kumpulan aturan seperti yang dijelaskan sebelumnya, Anda akan diarahkan ke halaman aturan kualitas data, yang menampilkan semua kumpulan aturan di akun Anda.

Untuk membuat pekerjaan profil termasuk kumpulan aturan

1. Pilih nama kumpulan aturan yang sebelumnya Anda buat untuk melihat detailnya.
2. Pilih Buat pekerjaan profil dengan ruleset.

   Nama Job terisi secara otomatis, tetapi Anda dapat mengubahnya sesuai kebutuhan.
3. Untuk contoh Job run, Anda dapat memilih untuk menjalankan seluruh kumpulan data atau sejumlah baris terbatas.

   Jika Anda memilih untuk menjalankan ukuran sampel terbatas, ketahuilah bahwa untuk aturan tertentu, hasilnya mungkin berbeda dibandingkan dengan kumpulan data lengkap.

4. Untuk pengaturan output Job, pilih lokasi S3 untuk output pekerjaan. Pilih folder apa pun di bucket Amazon S3 bernama yang dapat Anda akses. Jika Anda memasukkan nama folder untuk bucket ini yang tidak ada, folder ini akan dibuat.

   Setelah berhasil menyelesaikan pekerjaan profil, folder ini akan berisi profil laporan validasi aturan kualitas data dan data dalam format JSON.
5. Di bawah Aturan kualitas data, perhatikan kumpulan aturan Anda tercantum di bawah Nama set aturan kualitas data.
6. Di bawah Izin, pilih atau buat peran untuk memberikan DataBrew akses membaca dari lokasi input Amazon S3 dan tulis ke lokasi keluaran pekerjaan. Jika Anda belum memiliki peran yang siap, pilih Buat peran IAM baru.
7. Ubah pengaturan opsional lainnya seperti yang dijelaskan dalamMembuat dan bekerja dengan AWS Glue DataBrew lowongan kerja profil, jika diperlukan.
8. Pilih Buat dan jalankan pekerjaan.

Memeriksa hasil validasi untuk dan memperbarui aturan kualitas data

Setelah pekerjaan profil Anda selesai, Anda dapat melihat hasil validasi untuk aturan kualitas data Anda dan jika diperlukan memperbarui aturan Anda.

Untuk melihat data validasi untuk aturan kualitas data Anda

1. Di DataBrew konsol, pilih Lihat profil data. Melakukan hal ini akan menampilkan tab ikhtisar profil data untuk kumpulan data Anda.
2. Pilih tab Aturan kualitas data. Pada tab ini, Anda dapat melihat hasil untuk semua aturan kualitas data Anda.
3. Pilih aturan individual untuk detail lebih lanjut tentang aturan itu.

Untuk aturan apa pun yang gagal validasi, Anda dapat melakukan koreksi yang diperlukan.

Untuk memperbarui aturan kualitas data

1. Pada panel navigasi, pilih ATURAN DQ.
2. Di bawah Nama set aturan kualitas data, pilih kumpulan data yang berisi aturan yang ingin Anda edit.

3. Pilih aturan yang ingin Anda ubah, lalu pilih Edit.
4. Lakukan koreksi yang diperlukan, lalu pilih Perbarui aturan.
5. Jalankan kembali pekerjaannya. Ulangi proses ini sampai semua validasi berlalu.

Cek yang tersedia

Tabel berikut mencantumkan referensi untuk semua kondisi yang tersedia yang dapat digunakan dalam aturan Anda. Perhatikan bahwa kondisi agregat tidak dapat digabungkan dengan kondisi non-agregat dalam aturan yang sama.

> **Note**
>
> Untuk pengguna SDK, untuk menerapkan aturan yang sama ke beberapa kolom, gunakan ColumnSelectorsatribut Aturan dan tentukan kolom yang divalidasi menggunakan nama atau ekspresi reguler. Dalam hal ini, Anda harus menggunakan implisit CheckExpression. Misalnya, "`> :val`" untuk membandingkan nilai di setiap kolom yang dipilih dengan nilai yang diberikan. DataBrew menggunakan sintaks implisit untuk mendefinisikan FilterExpressiondalam kumpulan data dinamis. Jika Anda ingin menentukan kolom untuk setiap pemeriksaan satu per satu, jangan atur ColumnSelectorsatribut. Sebaliknya, berikan ekspresi eksplisit. Misalnya, "`:col > :val`" seperti CheckExpressiondalam Aturan.

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK
Kondisi kumpulan data agregat	Jumlah baris		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(ROWS_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK
	Jumlah kolom		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(COLUMNS_COUNT) == :val", "SubstitutionMap": {":val", "20"}`

	Baris duplikat		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(DUPLICATE_ROWS_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(DUPLICATE_ROWS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

Kondisi statistik kolom agregat	Nilai yang hilang		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(MISSING_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(MISSING_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai duplikat		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(DUPLICATE_VALUES_COUNT) < :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(DUPLICATE_VALUES_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai valid		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(VALID_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "10000"}`
				
				atau
				
				`"CheckExpression": "AGG(VALID_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "95"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai yang berbeda		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(DISTINCT_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "1000"}`
				
				atau
				
				`"CheckExpression": "AGG(DISTINCT_VALUES_PERCENTAGE) >= :val", "SubstitutionMap": {":val", "50"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai unik		Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(UNIQUE_VALUES_COUNT) > :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(UNIQUE_VALUES_PERCENTAGE) > :val", "SubstitutionMap": {":val", "20"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Outlier	Z-score ambang	Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_COUNT, :zscore_dev) < :val", "SubstitutionMap": {":zscore_dev": "4", ":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(Z_SCORE_OUTLIERS_PERCENTAGE) < :val", "SubstitutionMap": {":val", "5"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Statistik distribusi nilai	Nama statistik (lihat tabel berikutnya)	Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Lihat tabel berikutnya untuk kemungkinan `STAT_NAME` nilai

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Statistik numerik	Nama statistik (lihat tabel berikutnya)	Perbandingan numerik terhadap nilai kustom	`"CheckExpression": "AGG(<STAT_NAME>) < :val", "SubstitutionMap": {":val", "100"}`
				
				atau
				
				`"CheckExpression": "AGG(<STAT_NAME>, :param) < :val", "SubstitutionMap": {":param": "0.25", :val", "5"}`
				
				Note
				Lihat tabel berikutnya untuk kemungkinan `STAT_NAME` nilai

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK
Non agregat (menerima ambang batas)	Nilai persis		Perbandingan yang tepat terhadap daftar nilai	`"CheckExpression": ":col IN :list", "SubstitutionMap": {":col": "`size`", ":list": "[\"S\",\"M\",\"L\",\"XL\"]"}`
	Nilai tidak persis		Nilai seharusnya tidak sama persis dengan nilai apa pun dari daftar	`"CheckExpression": ":col NOT IN :list", "SubstitutionMap": {":col": "`domain`", ":list": "[\"GOV\",\"ORG\"]"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai string		Perbandingan string terhadap nilai kustom atau kolom string lainnya	`"CheckExpression": ":col STARTS_WITH :val", "SubstitutionMap": {":col": "`url`", ":val": "http"}`
				
				atau
				
				`"CheckExpression": ":col1 contains :col2", "SubstitutionMap": {":col1": "`url`", ":col2": "`company_name`"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai numerik		Perbandingan numerik terhadap nilai kustom atau kolom numerik lainnya	`"CheckExpression": ":col IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`APY`", ":val1": "0", ":val2": "10"}`
				
				atau
				
				`"CheckExpression": ":col1 <= :col2", "SubstitutionMap": {":col1": "`bank_rate`", ":col2": "`fed_rate`"}`

Jenis syarat	Pemeriksaan kualitas data	Parameter tambahan	Jenis perbandingan	Contoh sintaks SDK

	Nilai panjang string		Perbandingan numerik terhadap nilai kustom atau kolom numerik lainnya	`"CheckExpression": "length(:col) IS_BETWEEN :val1 and :val2", "SubstitutionMap": {":col": "`identifier`", ":val1": "8", ":val2": "12"}`
				
				atau
				
				`"CheckExpression": "length(:col1) <= :col2", "SubstitutionMap": {":col1": "`name`", ":col2": "`max_name_len`"}`

Perbandingan numerik

DataBrew mendukung operasi berikut untuk perbandingan numerik: Apakah sama (= =), Tidak sama (! =), Kurang dari (<), Kurang dari sama (< =), Lebih besar dari (>), Lebih besar dari sama (> =) dan Adalah antara (is_between:val1 dan:val2).

Perbandingan string

Perbandingan string berikut didukung: Dimulai dengan, Tidak dimulai dengan, Berakhir dengan, Tidak diakhiri dengan, Berisi, Tidak mengandung, Sama dengan, Tidak sama, Cocokkan, Tidak cocok.

Tabel berikut menampilkan statistik yang tersedia yang dapat Anda gunakan untuk statistik distribusi Nilai dan statistik numerik:

Pemeriksaan kualitas data	Nama statistik	Parameter tambahan	Sintaks SDK
Statistik distribusi nilai	Min		`"CheckExpression": "AGG(MAX) < :val", "SubstitutionMap": {":val", "100"}`
	Maks		`"CheckExpression": "AGG(MIN) > :val", "SubstitutionMap": {":val", "0"}`
	Median		`"CheckExpression": "AGG(MEDIAN) >= :val", "SubstitutionMap": {":val", "50"}`

Pemeriksaan kualitas data	Nama statistik	Parameter tambahan	Sintaks SDK
	Berarti		`"CheckExpression": "AGG(MEAN) <= :val", "SubstitutionMap": {":val", "10"}`
	Modus		`"CheckExpression": "AGG(MODE) > :val", "SubstitutionMap": {":val", "0"}`
	Standar deviasi		`"CheckExpression": "AGG(STANDARD_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Entropi		`"CheckExpression": "AGG(ENTROPY) > :val", "SubstitutionMap": {":val", "0"}`

Pemeriksaan kualitas data	Nama statistik	Parameter tambahan	Sintaks SDK
Statistik numerik	Jumlah		`"CheckExpression": "AGG(SUM) > :val", "SubstitutionMap": {":val", "0"}`
	Kurtosis		`"CheckExpression": "AGG(KURTOSIS) > :val", "SubstitutionMap": {":val", "0"}`
	Kemiringan		`"CheckExpression": "AGG(SKEWNESS) > :val", "SubstitutionMap": {":val", "0"}`
	Varians		`"CheckExpression": "AGG(VARIANCE) > :val", "SubstitutionMap": {":val", "0"}`

Pemeriksaan kualitas data	Nama statistik	Parameter tambahan	Sintaks SDK
	Deviasi absolut		`"CheckExpression": "AGG(MEDIAN_ABSOLUTE_DEVIATION) > :val", "SubstitutionMap": {":val", "0"}`
	Kuantil	Kuantil: salah satu dari '0,25', '0,5', '0,75'	`"CheckExpression": "AGG(QUANTILE, :pct) > :val", "SubstitutionMap": {":pct": "0.25", ":val", "0"}`

Membuat dan menggunakan AWS Glue DataBrew memproyeksikan

Pada tahun AWS Glue DataBrew, sebuah proyek adalah inti dari analisis data dan upaya transformasi Anda.

Saat Anda membuat proyek, Anda menyatukan dua komponen mendasar:

- Dataset, untuk menyediakan akses hanya-baca ke data sumber Anda. Untuk informasi selengkapnya, lihat Menghubungkan ke data dengan AWS Glue DataBrew.
- Resep, untuk menerapkan transformasi DataBrew data ke kumpulan data. Untuk informasi selengkapnya, lihat Membuat dan menggunakan AWS Glue DataBrew resep.

DataBrew Konsol menyajikan proyek Anda dalam antarmuka pengguna yang sangat interaktif dan intuitif. Ini mendorong Anda untuk bereksperimen dengan ratusan transformasi data, sehingga Anda dapat mempelajari cara kerjanya dan apa pengaruhnya terhadap data Anda.

Data yang Anda lihat dalam tampilan proyek adalah contoh kumpulan data Anda. Karena kumpulan data bisa sangat besar, dengan ribuan atau bahkan jutaan baris, menggunakan sampel membantu memastikan bahwa DataBrew konsol tetap responsif saat Anda mengubah data sampel dengan berbagai cara. Secara default, sampel terdiri dari 500 baris data pertama dari kumpulan data. Anda dapat memilih pengaturan yang berbeda untuk ukuran sampel, dan baris mana yang dipilih.

Saat Anda mengubah data sampel, DataBrew membantu Anda membangun dan menyempurnakan resep proyek—serangkaian transformasi langkah demi langkah yang Anda terapkan sejauh ini. Resep pekerjaan yang sedang berlangsung disimpan secara otomatis, sehingga Anda dapat meninggalkan tampilan proyek kapan saja, kembali nanti, dan mengambil tempat yang Anda tinggalkan.

Ketika resep Anda siap digunakan, Anda dapat mempublikasikannya. Menerbitkan resep membuatnya tersedia untuk subsistem DataBrew pekerjaan, di mana Anda dapat menerapkan resep ke seluruh kumpulan data Anda, atau membuat profil data ekstensif yang memungkinkan Anda memahami struktur, konten, dan karakteristik statistik data Anda.

Topik

- Membuat proyek

- Ikhtisar sesi DataBrew proyek
- Menghapus proyek

Membuat proyek

Gunakan prosedur berikut untuk membuat proyek.

Untuk membuat proyek

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol.
2. Pada panel navigasi, pilih PROJECTS. Kemudian pilih Buat proyek.
3. Masukkan nama untuk proyek Anda. Kemudian pilih resep untuk dilampirkan ke proyek Anda:
   - Pilih Buat resep baru jika Anda memulai dari awal. Melakukan hal ini menciptakan resep baru yang kosong dan menempelkannya ke proyek Anda.
   - Pilih Edit resep yang ada jika Anda memiliki resep yang diterbitkan sebelumnya yang ingin Anda gunakan untuk proyek ini. Jika resep saat ini dilampirkan ke proyek lain, atau memiliki pekerjaan yang ditentukan untuknya, maka Anda tidak dapat menggunakannya dalam proyek baru Anda. Pilih Jelajahi resep untuk melihat resep apa yang tersedia.
   - Pilih Impor langkah dari resep jika Anda memiliki resep yang sudah diterbitkan sebelumnya dan ingin mengimpor langkah-langkahnya, lalu lakukan hal berikut:
     1. Pilih Jelajahi resep untuk melihat resep apa yang tersedia.
     2. Pilih versi resep yang diterbitkan yang ingin Anda gunakan. Sebuah resep dapat memiliki beberapa versi, tergantung pada seberapa sering Anda menerbitkannya saat bekerja dalam tampilan proyek.
     3. Pilih Lihat langkah-langkah resep untuk memeriksa transformasi data dalam resep.
4. Setelah Anda memiliki resep, pilih kumpulan data yang ingin Anda gunakan di panel Pilih kumpulan data:
   - Kumpulan data saya — Pilih kumpulan data yang Anda buat sebelumnya. Untuk informasi lebih lanjut, lihat Membuat proyek.)
   - File sampel — Buat kumpulan data baru berdasarkan data sampel yang dikelola oleh AWS. Data sampel ini adalah cara yang bagus untuk mengeksplorasi apa yang DataBrew dapat dilakukan, tanpa harus memberikan data Anda sendiri. Pastikan untuk memasukkan nama untuk dataset Anda.

- Dataset baru — Buat dataset baru. Untuk informasi selengkapnya, lihat Membuat proyek.

5. Untuk izin Akses, pilih peran AWS Identity and Access Management(IAM) yang memungkinkan DataBrew untuk membaca dari lokasi input Amazon S3 Anda. Untuk lokasi S3 yang dimiliki oleh AWS akun Anda, Anda dapat memilih peran yang `AwsGlueDataBrewDataAccessRole` dikelola layanan. Melakukan hal ini memungkinkan DataBrew untuk mengakses sumber daya S3 yang Anda miliki.
6. Pada panel Sampling, Anda dapat menemukan opsi DataBrew untuk membuat sampel data dari kumpulan data Anda.

   Untuk Type, pilih bagaimana DataBrew seharusnya mendapatkan baris dari dataset Anda:

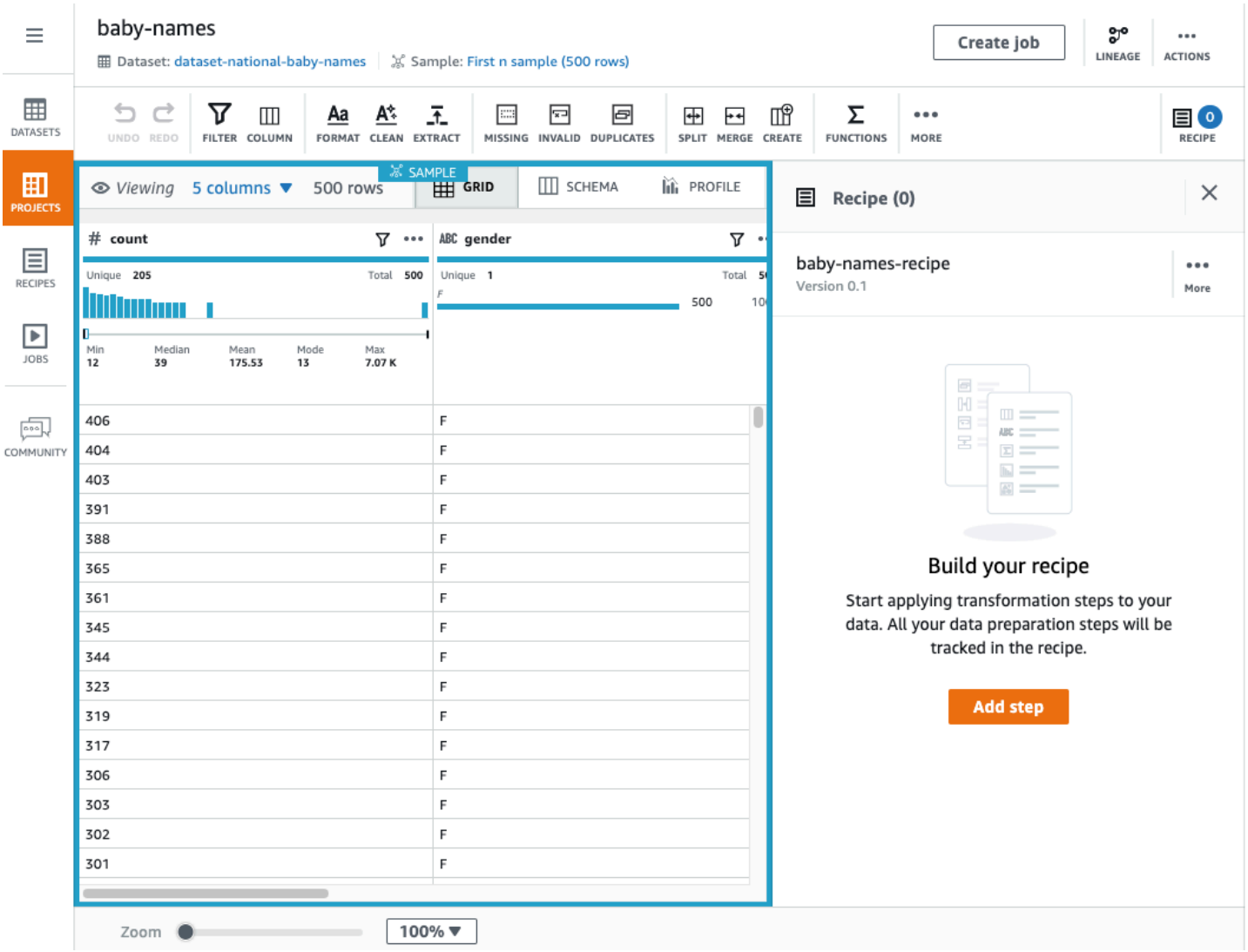
   - Gunakan baris n Pertama untuk membuat sampel berdasarkan baris pertama dalam kumpulan data.
   - Gunakan baris Acak untuk membuat sampel berdasarkan pemilihan baris acak dalam kumpulan data.
   - Pilih jumlah baris yang akan muncul dalam sampel: 500, 1.000, 2.500, atau ukuran sampel khusus, hingga maksimum 5.000 baris. Ukuran sampel yang lebih kecil memungkinkan DataBrew untuk melakukan transformasi lebih cepat, menghemat waktu Anda saat Anda mengembangkan resep Anda. Ukuran sampel yang lebih besar lebih akurat mencerminkan susunan data sumber yang mendasarinya. Namun, inisialisasi sesi proyek dan transformasi interaktif lebih lambat.
7. (Opsional) Pilih Tag untuk melampirkan tag ke kumpulan data Anda.

   Tag adalah label sederhana yang terdiri dari kunci yang ditentukan pengguna dan nilai opsional yang dapat membuatnya lebih mudah untuk mengelola, mencari, dan memfilter DataBrew proyek berdasarkan tujuan, pemilik, lingkungan, atau kriteria lainnya.
8. Ketika pengaturan seperti yang Anda inginkan, pilih Buat pekerjaan.

DataBrew membuat dataset baru jika diperlukan, membuat resep baru jika diperlukan, membangun sampel data, dan membuat sesi proyek interaktif. Proses ini bisa memakan waktu beberapa menit untuk menyelesaikannya. Ketika proyek siap digunakan, Anda dapat mulai bekerja dengan sampel data.

Ikhtisar sesi DataBrew proyek

Dalam sesi DataBrew proyek, Anda bekerja dalam ruang kerja interaktif.



Panel kiri menunjukkan tampilan data Anda saat ini. Panel kanan menunjukkan resep transformasi proyek, yang saat ini kosong.

Di sudut kanan atas grid data, ada tiga tab:`GRID`,, dan. `SCHEMA` `PROFILE` Memilih salah satu tab ini menampilkan tampilan yang sesuai di ruang kerja; tampilan ini dijelaskan selanjutnya.

Tampilan kisi

Tampilan grid adalah tampilan default, di mana sampel ditampilkan dalam format tabel. Gunakan prosedur berikut untuk panduan singkat tampilan kisi.

Untuk mengambil panduan tampilan grid

1. Mulailah dengan melihat seluruh ruang:

a. Gulir ke kiri dan kanan untuk melihat semua kolom.

b. Gulir ke atas dan ke bawah untuk melihat semua nilai data.

c. Gunakan kontrol zoom di bagian bawah ruang kerja untuk menyesuaikan tingkat perbesaran kisi.

2. Di kanan atas, lihat berapa banyak kolom sampel yang ditampilkan dan jumlah baris saat ini dalam sampel.

   Untuk mengubah kolom mana yang ditampilkan, pilih tautan N kolom (di mana N adalah jumlah kolom yang saat ini ditampilkan). Pilih kolom yang Anda inginkan, dan pilih Tampilkan kolom yang dipilih.

3. Sekarang Anda dapat mulai bereksperimen dengan DataBrew transformasi. Coba tindakan berikut ini:

   a. Dari toolbar transformasi, pilih Pilih Format, Ubah ke huruf besar.

   b. Untuk kolom Sumber, pilih kolom yang berisi data karakter.

   c. Biarkan pengaturan lainnya tetap default.

   d. Untuk melihat seperti apa data yang diubah nantinya, pilih Pratinjau perubahan. Kemudian, untuk menambahkan transformasi ini ke resep Anda, pilih Terapkan.

   Setiap kali Anda menerapkan transformasi data, DataBrew tambahkan ke salinan resep Anda yang berfungsi. Ini muncul di sisi kanan ruang kerja Anda.

4. Coba tindakan berikut ini:

   a. Dari bilah alat transformasi, pilih Buat, Berdasarkan fungsi.

   b. Untuk Pilih fungsi, pilih`SQUARE ROOT`.

   c. Untuk kolom Sumber, pilih kolom yang berisi data numerik.

   d. Biarkan pengaturan lain pada defaultnya,.

   e. Pilih Pratinjau perubahan untuk melihat seperti apa tampilan data yang diubah. Kemudian, untuk menambahkan transformasi ini ke resep Anda, pilih Terapkan.

5. Tutup panel resep di kanan atas dengan memilih RECIPE. Untuk memperluas panel resep, pilih RECIPE lagi.

Menerbitkan versi baru resep Anda

Saat Anda terus menerapkan transformasi, jumlah langkah dalam resep meningkat. Kapan saja, Anda dapat menerbitkan versi baru resep Anda. Menerbitkan resep membuatnya tersedia di tempat lain di DataBrew. Dengan melakukan ini, Anda dapat menjalankan pekerjaan resep untuk mengubah seluruh kumpulan data Anda, bukan hanya mengubah sampel data proyek.

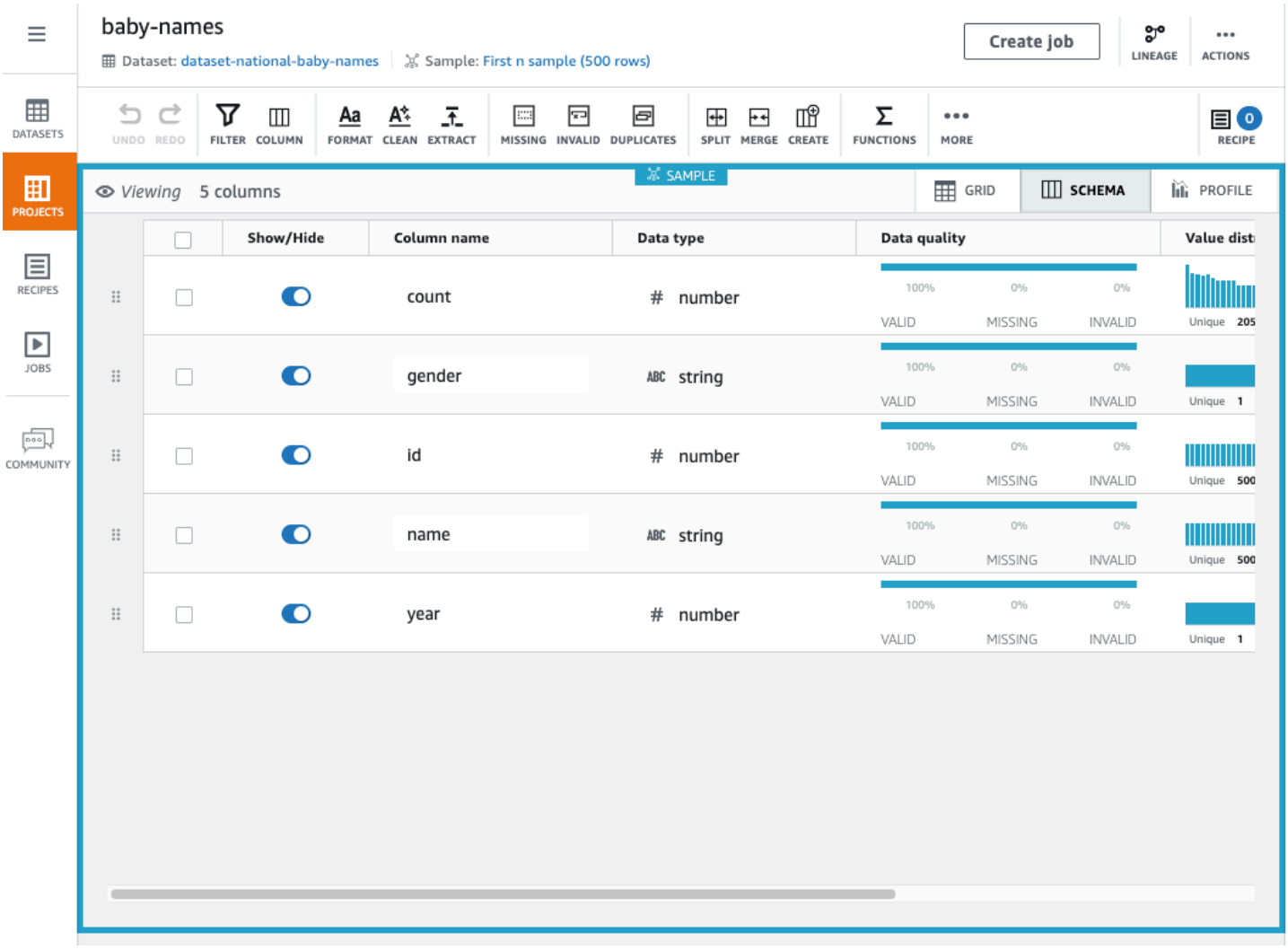
Menerbitkan resep juga mendorong pendekatan bertahap dan berulang untuk pengembangan resep: Anda dapat menerbitkan versi baru resep Anda saat Anda pergi, sehingga Anda dapat kembali ke versi resep “terakhir yang diketahui baik” jika diperlukan.

Untuk menerbitkan versi baru resep

- Di panel resep, pilih Publikasikan. Masukkan deskripsi untuk versi resep ini, dan pilih Publikasikan.

Tampilan skema

Jika Anda memilih tab SCHEMA, tampilan berubah, seperti yang ditunjukkan pada gambar berikut.



Dalam tampilan skema, Anda dapat melihat statistik tentang nilai data di setiap kolom.

Di kolom paling kiri, di samping Show/Hide, pilih salah satu kolom data. Panel detail Kolom muncul di sebelah kanan. Panel ini menunjukkan ringkasan statistik untuk nilai kolom.

Anda dapat mengganti nama kolom dengan memasukkan nama baru untuk nama Kolom.

Anda dapat mengatur ulang urutan kolom dengan menyeret dan menjatuhkan kolom.

Tampilan profil

Jika Anda memilih tab PROFIL, Anda dapat melihat informasi volumetrik terperinci tentang proyek Anda. Sebelum melakukannya, Anda menjalankan DataBrew pekerjaan untuk membuat profil.

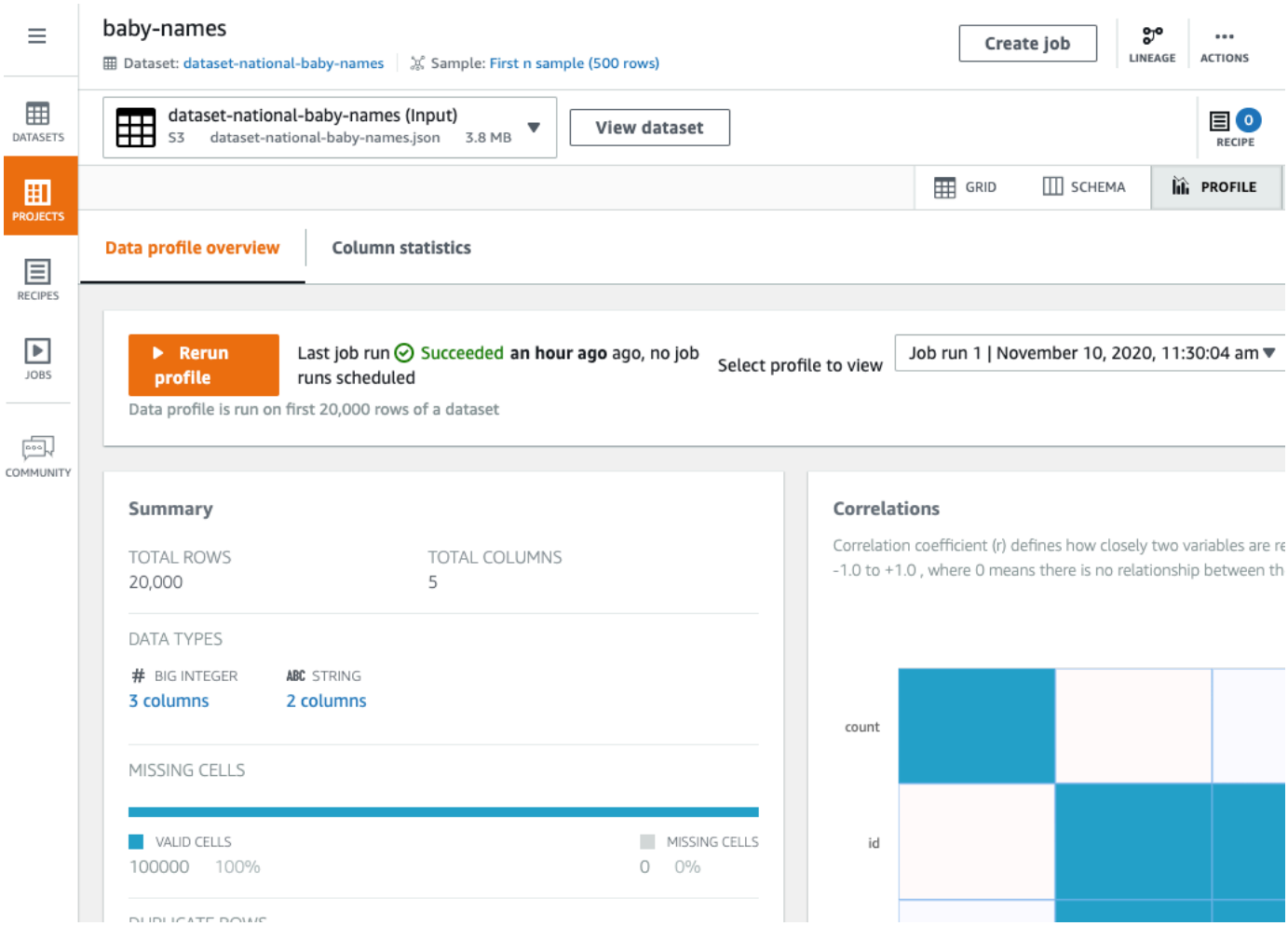
Untuk mengambil penelusuran tampilan profil

1. Pilih Buat pekerjaan, dan masukkan nama untuk pekerjaan Anda.
2. Untuk keluaran Job, pilih CSV untuk jenis file.
3. Temukan atau buat bucket dan folder Amazon S3 di AWS akun tempat Anda ingin hasil pekerjaan DataBrew ditulis:

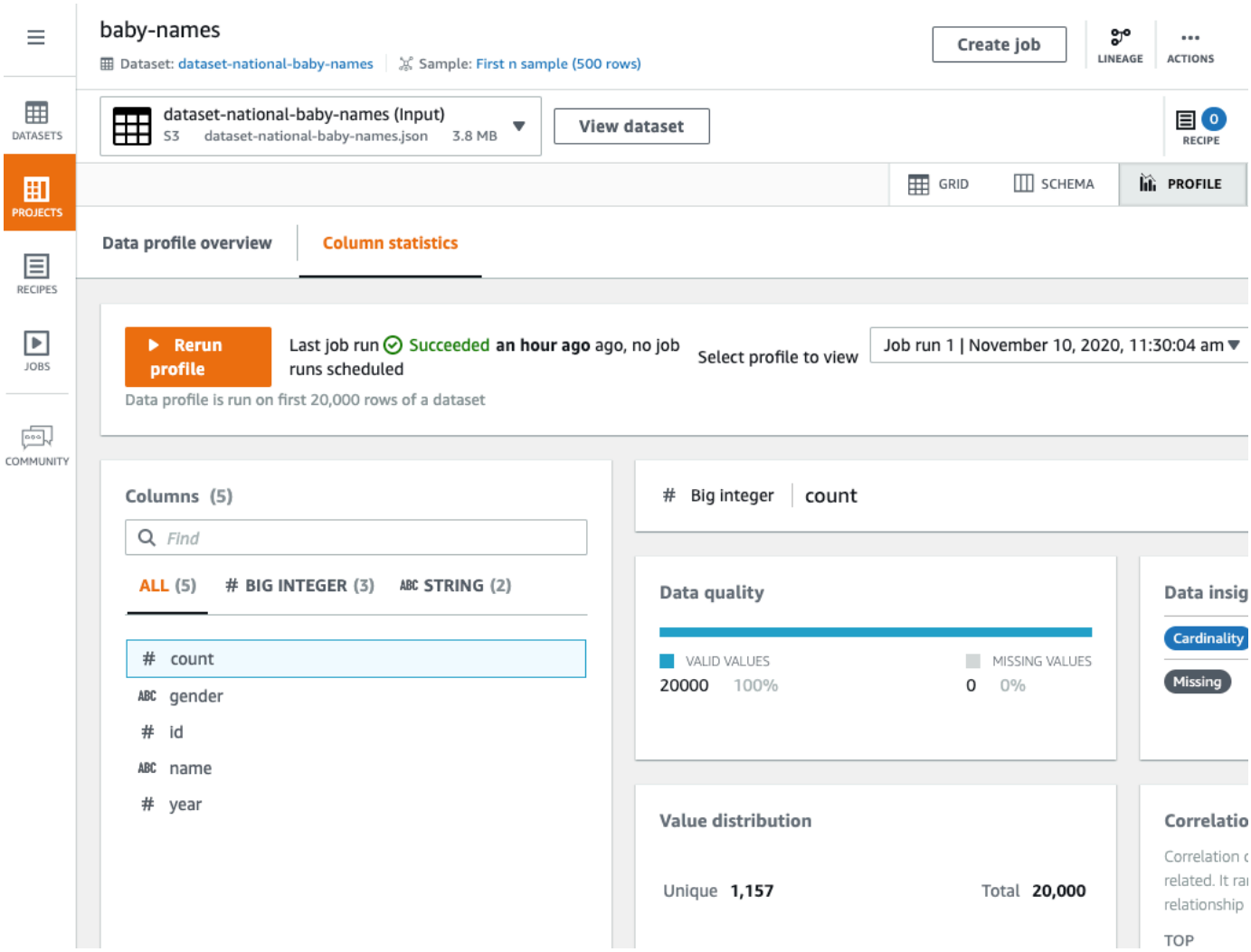
   - Jika Anda sudah memiliki bucket dan folder Amazon S3 ini, pilih Browse dan temukan. Pastikan Anda memiliki izin menulis untuk keduanya.
   - Jika Anda tidak memiliki bucket dan folder Amazon S3 ini, buat:
     1. Buka konsol Amazon S3 di. https://console.aws.amazon.com/s3/
     2. Jika Anda tidak memiliki bucket Amazon S3, pilih Buat ember. Untuk nama Bucket, masukkan nama unik untuk bucket baru Anda. Pilih Buat bucket.
     3. Dari daftar ember, pilih salah satu yang ingin Anda gunakan.
     4. Pilih Buat folder. Untuk nama Folder, masukkan`databrew-output`, dan pilih Buat folder.
4. Untuk izin Akses, pilih peran IAM yang memungkinkan DataBrew untuk menulis ke lokasi keluaran Amazon S3 Anda.

   Untuk lokasi S3 yang dimiliki oleh AWS akun Anda, Anda dapat memilih peran yang `AwsGlueDataBrewDataAccessRole` dikelola layanan. Melakukan hal ini memungkinkan DataBrew untuk mengakses sumber daya S3 yang Anda miliki.
5. Biarkan pengaturan lain pada defaultnya, dan pilih Buat dan jalankan pekerjaan.
6. Setelah pekerjaan berjalan hingga selesai, ruang kerja menampilkan ringkasan grafis dari profil data.

   Tab ikhtisar profil data menunjukkan ringkasan tingkat tinggi dari karakteristik data Anda, seperti yang ditunjukkan pada gambar berikut.



Tab statistik Kolom menunjukkan rincian kolom demi kolom dari nilai data:



Menghapus proyek

Jika Anda tidak lagi membutuhkan proyek, Anda dapat menghapusnya.

Untuk menghapus proyek

1. Pada panel navigasi, pilih PROJECTS.
2. Pilih proyek yang ingin Anda hapus, lalu untuk Tindakan, pilih Hapus. .

Membuat dan menggunakan AWS Glue DataBrew resep

Dalam DataBrew, resep adalah serangkaian langkah transformasi data. Anda dapat menerapkan langkah-langkah ini ke sampel data Anda, atau menerapkan resep yang sama ke kumpulan data.

Cara termudah untuk mengembangkan resep adalah dengan membuat DataBrew proyek, di mana Anda dapat bekerja secara interaktif dengan sampel data Anda—untuk informasi lebih lanjut, lihat. Membuat dan menggunakan AWS Glue DataBrew memproyeksikan Sebagai bagian dari alur kerja pembuatan proyek, resep baru (kosong) dibuat dan dilampirkan ke proyek. Anda kemudian dapat mulai membangun resep Anda dengan menambahkan transformasi data.

> **Note**
>
> Anda dapat memasukkan hingga 100 transformasi data dalam satu DataBrew resep.

Saat Anda melanjutkan dengan mengembangkan resep Anda, Anda dapat menyimpan pekerjaan Anda dengan menerbitkan resep. DataBrew memelihara daftar versi yang diterbitkan untuk resep Anda. Anda dapat menggunakan versi apa pun yang diterbitkan dalam pekerjaan resep, untuk menjalankan resep (dalam pekerjaan resep) untuk mengubah kumpulan data Anda. Anda juga dapat mengunduh salinan langkah-langkah resep, sehingga Anda dapat menggunakan kembali resep di proyek lain atau transformasi kumpulan data lainnya.

Anda juga dapat mengembangkan DataBrew resep secara terprogram, menggunakan AWS Command Line Interface(AWS CLI) atau salah satu SDK AWS. Dalam DataBrew API, transformasi dikenal sebagai tindakan resep.

> **Note**
>
> Dalam sesi DataBrew proyek interaktif, setiap transformasi data yang Anda terapkan menghasilkan panggilan ke DataBrew API. Panggilan API ini terjadi secara otomatis, tanpa Anda harus mengetahui detail di balik layar.

Bahkan jika Anda bukan seorang programmer, akan sangat membantu untuk memahami struktur resep dan bagaimana DataBrew mengatur tindakan resep.

Topik

- Menerbitkan versi resep baru
- Mendefinisikan struktur resep

Menerbitkan versi resep baru

Anda menerbitkan versi baru resep dalam sesi DataBrew proyek interaktif.

Untuk menerbitkan versi resep baru

1. Di panel resep, pilih Publikasikan.
2. Masukkan deskripsi untuk versi resep ini, dan pilih Publikasikan.

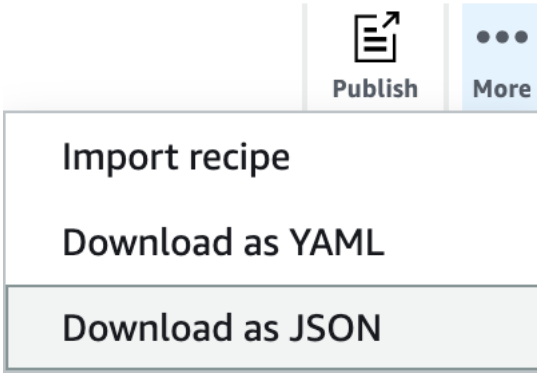
Anda dapat melihat semua resep yang diterbitkan, dan versinya, dengan memilih PROYEK dari panel navigasi.

Mendefinisikan struktur resep

Saat pertama kali membuat proyek menggunakan DataBrew konsol, Anda menentukan resep yang akan dikaitkan dengan proyek itu. Jika Anda tidak memiliki resep yang ada, konsol membuatnya untuk Anda.

Saat Anda bekerja dengan proyek di konsol, Anda menggunakan bilah alat transformasi untuk menerapkan tindakan ke data sampel dari kumpulan data Anda. Konsol menunjukkan langkah-langkah resep, dan urutan langkah-langkah itu, saat Anda terus membangun resep. Anda dapat mengulangi dan menyempurnakan resep sampai Anda puas dengan langkah-langkahnya.

DiMemulai dengan AWS Glue DataBrew, Anda membuat resep untuk mengubah kumpulan data permainan catur terkenal. Anda dapat mengunduh salinan langkah-langkah resep, dengan memilih Unduh sebagai JSON atau Unduh sebagai YAMAL seperti yang ditunjukkan pada gambar berikut.



File JSON yang diunduh berisi tindakan resep yang sesuai dengan transformasi yang Anda tambahkan ke resep Anda.

Resep baru tidak memiliki langkah apa pun. Anda dapat mewakili resep baru sebagai daftar JSON kosong, seperti yang ditunjukkan berikut.

```
[ ]
```

Berikut ini adalah contoh dari file tersebut, untuk`chess-project-recipe`. Daftar JSON berisi beberapa objek yang menjelaskan langkah-langkah resep. Setiap objek dalam daftar JSON terlampir dalam kurawal kurawal (). `{ }` Garis JSON dibatasi oleh koma.

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

Lebih mudah untuk melihat masing-masing bahwa setiap tindakan adalah baris individual jika kita hanya menambahkan baris baru untuk tindakan baru, seperti yang ditunjukkan berikut.

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

Tindakan dilakukan secara berurutan, dalam urutan yang sama seperti pada file:

- `REMOVE_VALUES`— Untuk menyaring semua game di mana peringkat pemain kurang dari 1.800, peringkat minimum yang diperlukan untuk menjadi pemain catur Kelas A. Ada dua kejadian dari tindakan ini — satu untuk menghapus pemain di sisi hitam yang tidak setidaknya pemain Kelas A, dan satu lagi untuk menghapus pemain di sisi putih yang tidak pada level ini.
- `GROUP_BY`— Untuk meringkas data. Dalam hal ini, GROUP_BY mengurutkan baris ke dalam grup berdasarkan nilai `winner` (`blackdanwhite`). Masing-masing kelompok tersebut kemudian dipecah lebih lanjut, mengurutkan baris menjadi subkelompok berdasarkan nilai `victory_status` (`mate`,, `resignoutoftime`, dan`draw`). Akhirnya, jumlah kejadian untuk setiap subkelompok dihitung. Ringkasan yang dihasilkan kemudian menggantikan sampel data asli.
- `REMOVE_VALUES`— Untuk menghapus hasil game yang diakhiri dengan`draw`.
- `REPLACE_TEXT`— Untuk memodifikasi nilai untuk`victory_status`. Ada tiga kejadian dari tindakan ini—masing-masing satu untuk`mate`,, dan. `resign` `oufoftime`

Dalam sesi DataBrew proyek interaktif, masing-masing `RecipeAction` sesuai dengan transformasi data yang Anda terapkan pada sampel data.

DataBrew menyediakan lebih dari 200 tindakan resep. Untuk informasi selengkapnya, lihat Langkah resep dan referensi fungsi.

Ketentuan penggunaan

Anda dapat menggunakan kondisi untuk mempersempit ruang lingkup tindakan resep. Kondisi digunakan dalam transformasi yang menyaring data—misalnya, menghapus baris yang tidak diinginkan berdasarkan nilai kolom tertentu.

Mari kita lihat lebih dekat tindakan resep dari`chess-project-recipe`.

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

```
    }
  ]
}
```

Transformasi ini membaca nilai-nilai dalam `black_rating` kolom. `ConditionExpressions`Daftar menentukan kriteria pemfilteran: Setiap baris yang memiliki `black_rating` nilai kurang dari 1.800 dihapus dari kumpulan data.

Transformasi tindak lanjut dalam resep melakukan hal yang sama, untuk`white_rating`. Dengan cara ini, data terbatas pada game di mana setiap pemain (hitam atau putih) dinilai di Kelas A atau lebih tinggi.

Berikut adalah contoh lain dari kondisi, diterapkan ke kolom data karakter.

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

Transformasi ini membaca nilai-nilai di `winner` kolom, mencari nilai `draw` dan menghapus baris tersebut. Dengan cara ini, data terbatas hanya pada game-game di mana ada pemenang yang jelas.

DataBrew mendukung kondisi berikut:

- `IS`— Nilai dalam kolom sama dengan nilai yang diberikan dalam kondisi.
- `IS_NOT`— Nilai di kolom tidak sama dengan nilai yang diberikan dalam kondisi.
- `IS_BETWEEN`— Nilai dalam kolom adalah antara `LESS_THAN_EQUAL` parameter `GREATER_THAN_EQUAL` dan.
- `CONTAINS`— Nilai string di kolom berisi nilai yang disediakan dalam kondisi.

- `NOT_CONTAINS`— Nilai dalam kolom tidak mengandung string karakter yang disediakan dalam kondisi.
- `STARTS_WITH`— Nilai di kolom dimulai dengan string karakter yang disediakan dalam kondisi.
- `NOT_STARTS_WITH`— Nilai di kolom tidak dimulai dengan string karakter yang disediakan dalam kondisi.
- `ENDS_WITH`— Nilai di kolom diakhiri dengan string karakter yang disediakan dalam kondisi.
- `NOT_ENDS_WITH`— Nilai di kolom tidak berakhir dengan string karakter yang disediakan dalam kondisi.
- `LESS_THAN`— Nilai dalam kolom kurang dari nilai yang diberikan dalam kondisi.
- `LESS_THAN_EQUAL`— Nilai dalam kolom kurang dari atau sama dengan nilai yang diberikan dalam kondisi.
- `GREATER_THAN`— Nilai dalam kolom lebih besar dari nilai yang diberikan dalam kondisi.
- `GREATER_THAN_EQUAL`— Nilai dalam kolom lebih besar dari atau sama dengan nilai yang diberikan dalam kondisi.
- `IS_INVALID`— Nilai di kolom memiliki tipe data yang salah.
- `IS_MISSING`— Tidak ada nilai di kolom.

Membuat, menjalankan, dan menjadwalkan AWS Glue DataBrew pekerjaan

AWS Glue DataBrew memiliki subsistem pekerjaan yang melayani dua tujuan:

1. Menerapkan resep transformasi data ke DataBrew kumpulan data. Anda melakukan ini dengan pekerjaan DataBrew resep.
2. Menganalisis kumpulan data untuk membuat profil data yang komprehensif. Anda melakukan ini dengan pekerjaan DataBrew profil.

Topik

- Membuat dan bekerja dengan AWS Glue DataBrew pekerjaan resep
- Membuat dan bekerja dengan AWS Glue DataBrew lowongan kerja profil

Membuat dan bekerja dengan AWS Glue DataBrew pekerjaan resep

Gunakan pekerjaan DataBrew resep untuk membersihkan dan menormalkan data dalam DataBrew kumpulan data dan tulis hasilnya ke lokasi keluaran pilihan Anda. Menjalankan pekerjaan resep tidak memengaruhi kumpulan data atau data sumber yang mendasarinya. Ketika pekerjaan berjalan, ia terhubung ke data sumber dengan cara read-only. Output pekerjaan ditulis ke lokasi keluaran yang Anda tentukan di Amazon S3, database JDBC AWS Glue Data Catalog, atau JDBC yang didukung.

Gunakan prosedur berikut untuk membuat pekerjaan DataBrew resep.

Untuk membuat pekerjaan resep

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol di https://console.aws.amazon.com/databrew/.
2. Pilih JOBS dari panel navigasi, pilih tab Pekerjaan resep, lalu pilih Buat pekerjaan.
3. Masukkan nama untuk pekerjaan Anda, lalu pilih Buat pekerjaan resep.
4. Untuk masukan Job, masukkan detail pekerjaan yang ingin Anda buat: nama kumpulan data yang akan diproses, dan resep yang akan digunakan.

Pekerjaan resep menggunakan DataBrew resep untuk mengubah kumpulan data. Untuk menggunakan resep, pastikan untuk mempublikasikannya terlebih dahulu.

5. Konfigurasikan pengaturan output pekerjaan Anda.

   Berikan tujuan untuk output pekerjaan Anda. Jika Anda tidak memiliki DataBrew koneksi yang dikonfigurasi untuk tujuan keluaran Anda, konfigurasikan terlebih dahulu pada tab DATASETS seperti yang dijelaskan di. Koneksi yang didukung untuk sumber data dan output Pilih salah satu tujuan output berikut:

   - Amazon S3, dengan atau tanpa dukungan AWS Glue Data Catalog
   - Amazon Redshift, dengan atau tanpa dukungan AWS Glue Data Catalog
   - JDBC
   - Tabel kepingan salju
   - Tabel database Amazon RDS dengan AWS Glue Data Catalog dukungan. Tabel database Amazon RDS mendukung mesin database berikut:
     - Amazon Aurora
     - MySQL
     - Oracle
     - PostgreSQL
     - Microsoft SQL Server
   - Amazon S3 dengan AWS Glue Data Catalog dukungan.

   Untuk AWS Glue Data Catalog output berdasarkan AWS Lake Formation, hanya DataBrew mendukung penggantian file yang ada. Dalam pendekatan ini, file diganti untuk menjaga izin Lake Formation yang ada tetap utuh untuk peran akses data Anda. Juga, DataBrew memberikan prioritas ke lokasi Amazon S3 dari tabel.AWS Glue Data Catalog Dengan demikian, Anda tidak dapat mengganti lokasi Amazon S3 saat membuat pekerjaan resep.

   Dalam beberapa kasus, lokasi Amazon S3 dalam output pekerjaan berbeda dari lokasi Amazon S3 di tabel Katalog Data. Dalam kasus ini, DataBrew perbarui definisi pekerjaan secara otomatis dengan lokasi Amazon S3 dari tabel katalog. Ini dilakukan ketika Anda memperbarui atau memulai pekerjaan yang ada.

6. Hanya untuk tujuan keluaran Amazon S3, Anda memiliki pilihan lebih lanjut:

a. Pilih salah satu format output data yang tersedia untuk Amazon S3, kompresi opsional, dan pembatas kustom opsional. Pembatas yang didukung untuk file output sama dengan yang untuk input: koma, titik dua, titik koma, pipa, tab, tanda sisipan, garis miring terbalik, dan spasi. Untuk detail pemformatan, lihat tabel berikut.

Format	Ekstensi file (tidak terkompresi)	Ekstensi file (terkompresi)
Comma-separated nilai	`.csv`	`.csv.snappy` , `.csv.gz`, `.csv.lz4`, `csv.bz2`, `.csv.deflate` , `csv.br`
Tab-separated nilai	`.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 Parket	Tidak didukung	`.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	Ekstensi file (tidak terkompresi)	Ekstensi file (terkompresi)
JSON (format JSON Lines saja)	`.json`	`.json.snappy` , `.json.gz`, `.json.lz4` , `json.bz2`, `.json.deflate` , `.json.br`
Tablo Hyper	Tidak didukung	Tidak berlaku

b. Pilih apakah akan menampilkan satu file atau beberapa file. Ada tiga opsi untuk output file dengan Amazon S3:

- Autogenerate file (disarankan) - Telah DataBrew menentukan jumlah file output yang optimal.
- Output file tunggal - Menyebabkan satu file output dihasilkan. Opsi ini dapat menghasilkan waktu eksekusi pekerjaan tambahan karena pasca-pemrosesan diperlukan.
- Beberapa output file - Apakah Anda menentukan jumlah file untuk output pekerjaan Anda. Nilai yang valid adalah 2-999. File yang lebih sedikit daripada yang Anda tentukan mungkin output jika partisi kolom digunakan atau jika jumlah baris dalam output lebih sedikit dari jumlah file yang Anda tentukan.

c. (Opsional) Pilih partisi kolom untuk output pekerjaan resep.

Partisi kolom menyediakan cara lain untuk mempartisi output pekerjaan resep Anda menjadi beberapa file. Partisi kolom dapat digunakan dengan output Amazon S3 baru atau yang sudah ada atau dengan keluaran Katalog Data Amazon S3 baru. Itu tidak dapat digunakan dengan tabel Data Catalog Amazon S3 yang ada. File output didasarkan pada nilai nama kolom yang Anda tentukan. Jika nama kolom yang Anda tentukan unik, jalur folder Amazon S3 yang dihasilkan didasarkan pada urutan nama kolom.

Untuk contoh partisi kolom, lihatContoh partisi kolom, berikut.

7. (Opsional) Pilih Aktifkan enkripsi untuk output pekerjaan untuk mengenkripsi output pekerjaan yang DataBrew menulis ke lokasi keluaran Anda, lalu pilih metode enkripsi:

- Gunakan SSE-S3 enkripsi — Output dienkripsi menggunakan enkripsi sisi server dengan kunci enkripsi yang dikelola Amazon S3.

- Use AWS Key Management Service(AWS KMS) - Output dienkripsi menggunakan.AWS KMS Untuk menggunakan opsi ini, pilih Nama Sumber Daya Amazon (ARN) dari AWS KMS kunci yang ingin Anda gunakan. Jika Anda tidak memiliki AWS KMS kunci, Anda dapat membuatnya dengan memilih Buat AWS KMS kunci.

8. Untuk izin Akses, pilih peran AWS Identity and Access Management(IAM) yang memungkinkan DataBrew untuk menulis ke lokasi keluaran Anda. Untuk lokasi yang dimiliki oleh AWS akun Anda, Anda dapat memilih peran yang `AwsGlueDataBrewDataAccessRole` dikelola layanan. Melakukan hal ini memungkinkan DataBrew untuk mengakses AWS sumber daya yang Anda miliki.
9. Pada panel Pengaturan pekerjaan lanjutan, Anda dapat memilih opsi lainnya untuk menjalankan pekerjaan Anda:

   - Jumlah maksimum unit — DataBrew memproses pekerjaan menggunakan beberapa node komputasi, berjalan secara paralel. Jumlah default node adalah 5. Jumlah node maksimum adalah 149.
   - Job timeout - Jika pekerjaan membutuhkan lebih dari jumlah menit yang Anda tetapkan di sini untuk dijalankan, itu gagal dengan kesalahan batas waktu. Nilai default adalah 2.880 menit, atau 48 jam.
   - Jumlah percobaan ulang — Jika pekerjaan gagal saat berjalan, DataBrew dapat mencoba menjalankannya lagi. Secara default, pekerjaan tidak dicoba lagi.
   - Aktifkan CloudWatch Log Amazon untuk pekerjaan - Memungkinkan DataBrew untuk mempublikasikan informasi diagnostik ke CloudWatch Log. Log ini dapat berguna untuk tujuan pemecahan masalah, atau untuk detail lebih lanjut tentang bagaimana pekerjaan diproses.

10. Untuk Jadwal pekerjaan, Anda dapat menerapkan jadwal DataBrew kerja sehingga pekerjaan Anda berjalan pada waktu tertentu, atau secara berulang. Untuk informasi selengkapnya, lihat Mengotomatiskan pekerjaan berjalan dengan jadwal.
11. Ketika pengaturan seperti yang Anda inginkan, pilih Buat pekerjaan. Atau, jika Anda ingin segera menjalankan pekerjaan, pilih Buat dan jalankan pekerjaan.

Anda dapat memantau kemajuan pekerjaan Anda dengan memeriksa statusnya saat pekerjaan sedang berjalan. Ketika pekerjaan berjalan selesai, status berubah menjadi Succeeded. Output pekerjaan sekarang tersedia di lokasi keluaran yang Anda pilih.

DataBrew menyimpan definisi pekerjaan Anda, sehingga Anda dapat menjalankan pekerjaan yang sama nanti. Untuk menjalankan kembali pekerjaan, pilih Jobs dari panel navigasi. Pilih pekerjaan yang ingin Anda kerjakan, lalu pilih Jalankan pekerjaan.

Contoh partisi kolom

Sebagai contoh partisi kolom, asumsikan bahwa Anda menentukan tiga kolom, setiap baris berisi salah satu dari dua nilai yang mungkin. `Dept`Kolom dapat memiliki nilai `Admin` atau`Eng`. `Staff-type`Kolom dapat memiliki nilai `Part-time` atau`Full-time`. `Location`Kolom dapat memiliki nilai `Office1` atau`Office2`. Bucket Amazon S3 untuk hasil pekerjaan Anda terlihat seperti berikut ini.

```
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
```

Mengotomatiskan pekerjaan berjalan dengan jadwal

Anda dapat menjalankan kembali DataBrew pekerjaan kapan saja dan juga mengotomatiskan DataBrew pekerjaan berjalan dengan jadwal.

Untuk menjalankan kembali pekerjaan DataBrew

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol di https://console.aws.amazon.com/databrew/.
2. Pada panel navigasi, pilih Jobs. Pilih pekerjaan yang ingin Anda jalankan, lalu pilih Jalankan pekerjaan.

Untuk menjalankan DataBrew pekerjaan pada waktu tertentu, atau secara berulang, buat jadwal DataBrew kerja. Anda kemudian dapat mengatur pekerjaan Anda untuk berjalan sesuai dengan jadwal.

Untuk membuat jadwal DataBrew kerja

1. Pada panel navigasi DataBrew konsol, pilih Jobs. Pilih tab Jadwal, dan pilih Tambahkan jadwal.
2. Masukkan nama untuk jadwal Anda, lalu pilih nilai untuk frekuensi Jalankan:
   - Berulang — Pilih seberapa sering Anda ingin pekerjaan berjalan (misalnya, setiap 12 jam). Kemudian pilih hari atau hari mana untuk menjalankan pekerjaan. Secara opsional, Anda dapat memasukkan waktu hari ketika pekerjaan berjalan.
   - Pada waktu tertentu — Masukkan waktu hari ketika Anda ingin pekerjaan berjalan. Kemudian pilih hari atau hari mana untuk menjalankan pekerjaan.
   - Masukkan CRON — Tentukan jadwal pekerjaan dengan memasukkan ekspresi cron yang valid. Untuk informasi selengkapnya, lihat Bekerja dengan ekspresi cron untuk pekerjaan resep.
3. Jika pengaturan sudah sesuai keinginan Anda, pilih Simpan.

Untuk mengaitkan pekerjaan dengan jadwal

1. Pada panel navigasi, pilih Jobs.
2. Pilih pekerjaan yang ingin Anda kerjakan, lalu untuk Tindakan, pilih Edit. .
3. Pada panel Jadwalkan pekerjaan, pilih Jadwal asosiasi. Pilih nama jadwal yang ingin Anda gunakan.
4. Jika pengaturan sudah sesuai keinginan Anda, pilih Simpan.

Bekerja dengan ekspresi cron untuk pekerjaan resep

Ekspresi cron memiliki enam bidang yang diperlukan, yang dipisahkan oleh spasi putih. Sintaksnya adalah sebagai berikut.

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

Dalam sintaks sebelumnya, nilai dan wildcard berikut digunakan untuk bidang yang ditunjukkan.

Bidang	Nilai-nilai	Wildcard
Menit	0–59	, - * /
Jam	0–23	, - * /
Day-of-month	1–31	, - * ? / L W
Bulan	1—12 atau JAN-DEC	, - * /
Day-of-week	1—7 atau SUN-SAT	, - * ? / L
Tahun	1970–2199	, - * /

Gunakan wildcard ini sebagai berikut:

- Wildcard , (koma) mencakup nilai tambahan. Di `Month` lapangan, `JAN,FEB,MAR` termasuk Januari, Februari, dan Maret.
- Wildcard - (en dash) menentukan rentang. Di `Day` lapangan, 1-15 termasuk hari 1 hingga 15 dari bulan yang ditentukan.
- Wildcard * (bintang) mencakup semua nilai di bidang. Di `Hours` lapangan, * termasuk setiap jam.
- Wildcard / (garis miring) menentukan kenaikan. Di `Minutes` lapangan, Anda dapat masuk **`1/10`** untuk menentukan setiap menit ke-10, mulai dari menit pertama jam (misalnya, menit ke-11, 21, dan 31).
- Wildcard ? (tanda tanya) menentukan satu atau yang lain. Misalnya, anggaplah di `Day-of-month` bidang yang Anda masukkan 7. Jika Anda tidak peduli hari apa dalam minggu ketujuh, Anda kemudian dapat masuk? di `Day-of-week` lapangan.
- Wildcard L di `Day-of-week` bidang `Day-of-month` or menentukan hari terakhir bulan atau minggu.
- Wildcard W di kolom `Day-of-month` menentukan hari kerja. Di kolom `Day-of-month`, `3W` menentukan hari kerja yang paling dekat dengan pekan ketiga di bulan itu.

Bidang dan nilai ini memiliki batasan berikut:

- Anda tidak dapat menentukan kolom `Day-of-month` dan `Day-of-week` dalam ekspresi cron yang sama. Jika Anda menentukan sebuah nilai di salah satu kolom, maka Anda harus menggunakan ? (tanda tanya) di kolom yang lain.
- Ekspresi cron yang mengarah ke kecepatan lebih cepat dari 5 menit tidak didukung.

Anda dapat membuat jadwal, Anda dapat menggunakan contoh cron berikut.

Menit	Jam	Hari dalam sebulan	Bulan	Hari dalam seminggu	Tahun	Arti
0	10	*	*	?	*	Jalankan pukul 10:00 pagi (UTC) setiap hari
15	12	*	*	?	*	Jalankan pada pukul 12:15 malam (UTC) setiap hari
0	18	?	*	MON-FRI	*	Jalankan pada pukul 6:00 sore (UTC) setiap Senin hingga Jumat
0	8	1	*	?	*	Jalankan pukul 8:00 pagi (UTC) setiap hari pertama

Menit	Jam	Hari dalam sebulan	Bulan	Hari dalam seminggu	Tahun	Arti
						setiap bulan
0/15	*	*	*	?	*	Jalankan setiap 15 menit
0/10	*	?	*	MON-FRI	*	Jalankan setiap 10 menit Senin hingga Jumat
0/5	8–17	?	*	MON-FRI	*	Jalankan setiap 5 menit Senin hingga Jumat antara pukul 8:00 pagi sampai pukul 5:55 sore (UTC)

Misalnya, Anda dapat menggunakan ekspresi cron berikut untuk menjalankan pekerjaan setiap hari pada pukul 12:15 UTC.

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

Menghapus pekerjaan dan jadwal pekerjaan

Jika Anda tidak lagi membutuhkan pekerjaan atau jadwal kerja, Anda dapat menghapusnya.

Untuk menghapus pekerjaan

1. Pada panel navigasi, pilih Jobs.
2. Pilih pekerjaan yang ingin Anda hapus, lalu untuk Tindakan, pilih Hapus. .

Untuk menghapus jadwal kerja

1. Pada panel navigasi, pilih Pekerjaan, lalu pilih tab Jadwal.
2. Pilih jadwal yang ingin Anda hapus, lalu untuk Tindakan, pilih Hapus. .

Membuat dan bekerja dengan AWS Glue DataBrew lowongan kerja profil

Pekerjaan profil menjalankan serangkaian evaluasi pada kumpulan data dan menampilkan hasilnya ke Amazon S3. Informasi yang dikumpulkan oleh profil data membantu Anda memahami kumpulan data Anda dan memutuskan jenis langkah persiapan data yang mungkin ingin Anda jalankan dalam pekerjaan resep Anda.

Cara termudah untuk menjalankan pekerjaan profil adalah menggunakan DataBrew pengaturan default. Anda dapat mengonfigurasi pekerjaan profil Anda sebelum menjalankannya sehingga hanya mengembalikan informasi yang Anda inginkan.

Gunakan prosedur berikut untuk membuat pekerjaan DataBrew profil.

Untuk membuat pekerjaan profil

1. Masuk ke Konsol Manajemen AWS dan buka DataBrew konsol di https://console.aws.amazon.com/databrew/.
2. Pilih JOBS dari panel navigasi, pilih tab Profile jobs, lalu pilih Create job.
3. Masukkan nama untuk pekerjaan Anda, lalu pilih Buat pekerjaan profil.
4. Untuk masukan Job, berikan nama dataset yang akan diprofilkan.
5. (Opsional) Konfigurasikan hal berikut pada panel Konfigurasi profil data:

- Konfigurasi tingkat kumpulan data — Konfigurasikan detail pekerjaan profil Anda untuk semua kolom dalam kumpulan data Anda.

  Secara opsional, Anda dapat mengaktifkan kemampuan untuk mendeteksi dan menghitung baris duplikat dalam kumpulan data. Anda juga dapat memilih Aktifkan matriks korelasi dan pilih kolom untuk melihat seberapa dekat nilai-nilai dalam beberapa kolom terkait. Untuk detail statistik yang dapat Anda konfigurasikan di tingkat kumpulan data, lihatStatistik yang dapat dikonfigurasi pada tingkat dataset. Anda dapat mengonfigurasi statistik di DataBrew konsol, atau menggunakan DataBrew API atau AWS SDK.

- Konfigurasi tingkat kolom - Menggunakan pengaturan konfigurasi profil default, Anda dapat memilih kolom yang akan disertakan dalam pekerjaan profil Anda. Gunakan Tambahkan penggantian konfigurasi untuk memilih kolom yang membatasi jumlah statistik yang dikumpulkan, atau ganti konfigurasi default statistik tertentu. Untuk detail statistik yang dapat Anda konfigurasikan di tingkat kolom, lihatStatistik yang dapat dikonfigurasi di tingkat kolom. Anda dapat mengonfigurasi statistik di DataBrew konsol, atau menggunakan DataBrew API atau AWS SDK.

  Pastikan bahwa setiap penggantian konfigurasi yang Anda tentukan berlaku untuk kolom yang Anda sertakan dalam pekerjaan profil Anda. Jika ada konflik antara penggantian berbeda yang Anda konfigurasikan untuk kolom, penggantian konflik terakhir memiliki prioritas.

6. (Opsional) Anda dapat membuat aturan kualitas Data dan menerapkan aturan tambahan yang terkait dengan kumpulan data ini atau menghapus yang sudah diterapkan. Untuk informasi selengkapnya tentang validasi kualitas data, lihatMemvalidasi kualitas data di AWS Glue DataBrew.
7. Pada panel Pengaturan pekerjaan lanjutan, Anda dapat memilih opsi lainnya untuk menjalankan pekerjaan Anda:

   - Jumlah maksimum unit — DataBrew memproses pekerjaan menggunakan beberapa node komputasi, berjalan secara paralel. Jumlah default node adalah 5. Jumlah node maksimum adalah 149.
   - Job timeout - Jika pekerjaan membutuhkan lebih dari jumlah menit yang Anda tetapkan di sini untuk dijalankan, itu gagal dengan kesalahan batas waktu. Nilai default adalah 2.880 menit, atau 48 jam.
   - Jumlah percobaan ulang — Jika pekerjaan gagal saat berjalan, DataBrew dapat mencoba menjalankannya lagi. Secara default, pekerjaan tidak dicoba lagi.

- Aktifkan CloudWatch Log Amazon untuk pekerjaan - Memungkinkan DataBrew untuk mempublikasikan informasi diagnostik ke CloudWatch Log. Log ini dapat berguna untuk tujuan pemecahan masalah, atau untuk detail lebih lanjut tentang bagaimana pekerjaan diproses.

8. Untuk Jadwal Terkait, Anda dapat menerapkan jadwal kerja sehingga pekerjaan Anda berjalan pada waktu tertentu, atau secara berulang. DataBrew Untuk informasi selengkapnya, lihat Mengotomatiskan pekerjaan berjalan dengan jadwal.
9. Ketika pengaturan seperti yang Anda inginkan, pilih Buat pekerjaan. Atau, jika Anda ingin segera menjalankan pekerjaan, pilih Buat dan jalankan pekerjaan.

Membangun konfigurasi pekerjaan profil secara terprogram di AWS Glue DataBrew

Di bagian ini, Anda dapat menemukan deskripsi langkah dan fungsi pekerjaan profil yang dapat Anda gunakan secara terprogram. Anda dapat menggunakannya baik dari AWS Command Line Interface(AWS CLI) atau dengan menggunakan salah satu AWS SDK.

Dalam pekerjaan profil, Anda dapat menyesuaikan konfigurasi untuk mengontrol cara DataBrew mengevaluasi kumpulan data Anda. Anda dapat menerapkan konfigurasi ke kumpulan data atau menerapkannya ke kolom tertentu. Anda dapat membangun konfigurasi saat membuat pekerjaan profil, dan kemudian memperbaruinya kapan saja.

Struktur konfigurasi profil mencakup empat bagian:

- ProfileColumns bagian
- DatasetStatisticsConfiguration bagian
- ColumnStatisticsConfigurations bagian
- EntityDetectorConfiguration bagian untuk mengkonfigurasi PII

Berikut adalah contohnya.

```
{
    "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 bagian

Di `ProfileColumns` bagian struktur Anda, atur kolom dari kumpulan data yang ingin Anda evaluasi dalam pekerjaan profil Anda. `ProfileColumns`adalah daftar pemilih kolom (`Selectors`). Anda dapat menentukan nama kolom atau ekspresi reguler dalam pemilih kolom. Berikut contohnya.

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

Kapan `ProfileColumns` ditentukan, hanya kolom yang namanya cocok dengan nama atau ekspresi reguler `ProfileColumns` yang disertakan dalam pekerjaan profil. Jika pekerjaan profil tidak mendukung tipe data kolom yang dipilih, DataBrew lewati kolom yang dipilih selama pekerjaan dijalankan.

Jika tidak ProfileColumns ditentukan, pekerjaan profil mengevaluasi semua kolom yang didukung. Kolom yang didukung adalah kolom yang berisi data tipe data yang didukung: `ByteTypeShortType`,`IntegerType`,`LongType`,`FloatType`,`DoubleType`,,`String`, atau`Boolean`.

DatasetStatisticsConfiguration bagian

Di `DatasetStatisticsConfiguration` bagian struktur Anda, Anda dapat membangun konfigurasi untuk evaluasi antar kolom. Konfigurasi termasuk `IncludedStatistics` dan`Overrides`. Berikut contohnya.

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

Anda dapat memilih evaluasi yang ingin Anda miliki dengan menambahkan nama evaluasi. `IncludedStatistics` Berikut contohnya.

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

Saat Anda menentukan`IncludedStatistics`, hanya evaluasi dalam daftar yang disertakan dalam pekerjaan profil. Jika tidak `IncludedStatistics` ditentukan, pekerjaan profil menjalankan semua evaluasi yang didukung dengan pengaturan default. Anda dapat mengecualikan semua evaluasi dengan menambahkan NONE ke`IncludedStatistics`. Berikut contohnya.

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

Statistik yang dapat dikonfigurasi pada tingkat dataset

Di `DatasetStatisticsConfiguration` bagian struktur Anda, pekerjaan profil mendukung evaluasi yang ditunjukkan pada tabel berikut.

Nama statistik	Deskripsi	Tipe data yang didukung	Status default	Atribut hasil profil	Jenis hasil profil
DUPLIKATE_ROWS_COUNT	Hitungan baris duplikat dalam kumpulan data	all	Aktifkan	duplikat RowsCoun	Int
KORELASI	Koefisien Korelasi Pearson antara dua kolom	number	Aktifkan	korelasi (di setiap kolom yang dipilih)	Objek

Di`IncludedStatistics`, Anda dapat mengganti setelan default setiap evaluasi dengan menambahkan penggantian. Setiap penggantian mencakup nama evaluasi tertentu dan peta parameter.

Di`DatasetStatisticsConfiguration`, pekerjaan profil mendukung `CORRELATION` penggantian. Penggantian ini menghitung Koefisien Korelasi Pearson antara dua kolom dari daftar kolom yang

dipilih. Pengaturan default adalah memilih 10 kolom numerik pertama. Anda dapat menentukan sejumlah kolom atau daftar pemilih kolom untuk mengganti pengaturan default.

`CORRELATION`mengambil parameter ini:

- `columnNumber`— Jumlah kolom numerik. Pekerjaan profil memilih n kolom pertama dari kumpulan data. Nilai ini harus lebih besar dari 1. Gunakan `"ALL"` untuk memilih semua kolom numerik.
- `columnSelectors:`— Daftar pemilih kolom. Setiap pemilih dapat memiliki nama kolom atau ekspresi reguler.

Berikut contohnya.

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

ColumnStatisticsConfigurations bagian

Di `ColumnStatisticsConfigurations` bagian struktur Anda, Anda dapat membangun konfigurasi untuk kolom tertentu. `ColumnStatisticsConfigurations`adalah daftar `ColumnStatisticsConfiguration` pengaturan. Di`ColumnStatisticsConfiguration`, ada`Selectors`, daftar pemilih kolom, dan `Statistics` untuk konfigurasi statistik. Berikut contohnya.

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

```
    }
}
```

`Selectors`adalah daftar pemilih kolom. Seperti halnya`ProfileColumns`, Anda dapat menentukan nama kolom atau ekspresi reguler di setiap pemilih kolom. Saat Anda menentukan`Selectors`, konfigurasi kolom diterapkan ke kolom yang cocok dengan pemilih kolom mana pun. `Selectors` Jika tidak, konfigurasi diterapkan ke semua kolom yang didukung.

Di`Statistics`, Anda dapat mengganti pengaturan kolom yang dipilih. Seperti halnya`DatasetStatisticsConfiguration`, `Statistics` memiliki `IncludedStatistics` dan`Overrides`.

Untuk memilih evaluasi yang Anda inginkan, tambahkan nama evaluasi ke`IncludedStatistics`.

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

Saat Anda menentukan`IncludedStatistics`, hanya evaluasi dalam daftar yang disertakan dalam pekerjaan profil. Jika tidak, pekerjaan profil menjalankan semua evaluasi yang didukung dengan pengaturan default.

Anda dapat mengecualikan semua evaluasi dengan menambahkan `NONE` ke`IncludedStatistics`.

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

Dalam beberapa kasus, mungkin ada beberapa konfigurasi yang berbeda `ColumnStatisticsConfigurations` `IncludedStatistics` yang dapat Anda terapkan ke kolom yang sama. Dalam kasus ini, pekerjaan profil memilih konfigurasi terakhir `ColumnStatisticsConfigurations` dan menerapkannya `IncludedStatistics` ke kolom yang dipilih. Konfigurasi baru mengesampingkan konfigurasi yang lebih lama.

Statistik yang dapat dikonfigurasi di tingkat kolom

Di`ColumnStatisticsConfigurations`, pekerjaan profil mendukung evaluasi yang ditunjukkan pada tabel berikut.

Tipe data yang didukung `number` dalam tabel ini berarti bahwa tipe data atribut adalah salah satu dari berikut:`ByteType`,`ShortType`,`IntegerType`,`LongType`,`FloatType`, atau`DoubleType`.

Nama statistik	Deskripsi	Tipe data yang didukung	Status default	Atribut hasil profil	Jenis hasil profil
–	Nama kolomnya.	all	–	name	string
–	Tipe data kolom.	all	–	jenis	string
DISTINCT_VALUES_COUNT	Jumlah nilai yang berbeda. Nilai yang berbeda adalah nilai yang muncul setidaknya sekali.	number/boolean/string	Diaktifkan	berbedaValuesCount	Int
ENTROPI	Entropi (teori informasi).	number/boolean/string	Diaktifkan	entropi	Ganda
INTER_QUARTILE_RANGE	Kisaran antara 25 persen dan 75 persen dari angka.	number	Diaktifkan	InterQuartileRange	Ganda
KURTOSIS	Kurtosis kolom.	number	Diaktifkan	kurtosis	Ganda
MAX	Nilai maksimum di kolom.	number/string panjang	Diaktifkan	max	Int/Double
MAXIMUM_VALUES	Daftar nilai maksimum di kolom dan hitungannya.	number	Diaktifkan	MaximumVlues	Daftar
MEAN	Nilai rata-rata nilai di kolom.	number/string panjang	Diaktifkan	kejam	Ganda
MEDIAN	Median nilai di kolom.	number/string panjang	Diaktifkan	median	Ganda
MEDIAN_ABSOLUTE_DEVIASI	Median perbedaan absolut antara setiap titik data	number	Diaktifkan	medianAbsoluteDeviation	Ganda

Nama statistik	Deskripsi	Tipe data yang didukung	Status default	Atribut hasil profil	Jenis hasil profil
	dan median kolom numerik.				
MIN	Nilai minimum di kolom.	number/string panjang	Diaktifkan	min	Int/Double
MINIMUM_VALUES	Daftar nilai minimum di kolom dan hitungannya.	number	Diaktifkan	MinimumVa lues	Daftar
HILANG_VALUES_COUNT	Jumlah nilai yang hilang di kolom. String nol dan kosong dianggap hilang.	all	Diaktifkan	hilang ValuesCount	Int
MODE	Nilai yang paling sering terjadi di kolom. Jika beberapa nilai sering muncul, mode adalah salah satu nilai tersebut.	number/string panjang	Diaktifkan	Mode	Int/Double
MOST_COMMON_VALUES	Daftar nilai yang paling umum di kolom.	number/boolean/string	Diaktifkan	paling CommonVa ues	Daftar

Nama statistik	Deskripsi	Tipe data yang didukung	Status default	Atribut hasil profil	Jenis hasil profil
OUTLIER_DETECTION	Mendeteksi outlier di kolom dengan algoritma Z_score. Hitung jumlah outlier dan ekstrak daftar sampel dari outlier yang terdeteksi.	number/string panjang	Diaktifkan	zScoreOutliersCount, zScoreOutliersSample	Int/List
PERSENTIL	Nilai persentil kolom numerik (5%, 25%, 75%, 95%).	number	Diaktifkan	persentil5, persentil25, persentil75, persentil95	Ganda
RANGE	Rentang nilai di kolom.	number	Diaktifkan	jangkauan	Int/Double
KEMIRINGAN	Kemiringan nilai di kolom.	number	Diaktifkan	kemiringan	Ganda
STANDARD_DEVIASI	Deviasi standar sampel yang tidak bias dari nilai di kolom.	number/string panjang	Diaktifkan	StandarDeviasi	Ganda
JUMLAH	Jumlah nilai di kolom.	number	Diaktifkan	sum	Int/Double

Nama statistik	Deskripsi	Tipe data yang didukung	Status default	Atribut hasil profil	Jenis hasil profil
UNIK_VALUES_COUNT	Jumlah nilai unik. Nilai unik berarti bahwa nilai hanya muncul sekali.	number/boolean/string	Diaktifkan	uniqueValuesCount	Int
VALUE_DISTRIBUTION	Ukur distribusi nilai dalam kolom berdasarkan rentang.	number/string panjang	Diaktifkan	DistribusiNilai	Daftar
PERBEDAAN	Varians nilai di kolom.	number	Diaktifkan	perbedaan	Ganda
Z_SCORE_DISTRIBUTION	Ukur distribusi nilai z-skor titik data berdasarkan rentang.	number	Diaktifkan	zScoreDistribution	Daftar
ZEROS_COUNT	Jumlah nol (0s) di kolom.	number	Diaktifkan	ZerosCount	Int

Di`IncludedStatistics`, Anda dapat mengganti setiap parameter default evaluasi dengan menambahkan override. Setiap penggantian mencakup nama evaluasi tertentu dan peta parameter.

Parameter untuk ColumnStatisticsConfigurations kolom

Di`ColumnStatisticsConfigurations`, pekerjaan profil mendukung parameter berikut.

Dalam beberapa kasus, mungkin ada beberapa konfigurasi yang berbeda `ColumnStatisticsConfigurations` `IncludedStatistics` yang dapat Anda terapkan ke kolom yang sama. Dalam kasus ini, pekerjaan profil memilih konfigurasi terakhir `ColumnStatisticsConfigurations` dan menerapkannya `IncludedStatistics` ke kolom yang dipilih. Konfigurasi baru mengesampingkan konfigurasi yang lebih lama.

MAXIMUM_VALUES

Daftar nilai maksimum di kolom numerik dan jumlahnya. Ukuran daftar default adalah 5. Anda dapat mengganti ukuran daftar dengan menentukan nilai untuk. `sampleSize`

Pengaturan

`sampleSize`— Ukuran daftar yang mencakup jumlah maksimum dan jumlah nilai dalam kolom numerik. Nilai ini harus lebih besar dari 0. Gunakan `"ALL"` untuk daftar semua nilai.

Contoh

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

MINIMUM_VALUES

Daftar nilai minimum di kolom numerik dan jumlahnya. Ukuran daftar default adalah 5. Anda dapat mengganti ukuran daftar dengan menentukan nilai untuk. `sampleSize`

Pengaturan

`sampleSize`— Ukuran daftar yang mencakup jumlah maksimum dan jumlah nilai dalam kolom numerik. Nilai ini harus lebih besar dari 0. Gunakan `"ALL"` untuk daftar semua nilai.

Contoh

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

MOST_COMMON_VALUES

Daftar nilai yang paling umum di kolom dan jumlahnya. Ukuran daftar default adalah 50. Anda dapat mengganti ukuran daftar dengan menentukan nilai untuk. `sampleSize`

Pengaturan

`sampleSize`— Ukuran daftar yang mencakup jumlah maksimum dan jumlah nilai dalam kolom numerik. Nilai ini harus lebih besar dari 0. Gunakan `"ALL"` untuk daftar semua nilai.

Contoh

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

OUTLIER_DETECTION

Mendeteksi outlier di kolom numerik atau kolom string (berdasarkan panjang string) dengan algoritma Z_score.

Pekerjaan profil Anda menghitung jumlah outlier dan menghasilkan daftar sampel outlier dan z-score mereka. Daftar sampel diurutkan berdasarkan nilai absolut skor-z. Ukuran daftar default adalah 50.

Algoritma Z_Score mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi. Ambang batas outlier default adalah 3.

Anda dapat memberikan satu ambang batas lagi, ambang batas ringan, untuk mendapatkan informasi lebih lanjut. Ambang batas ringan Anda harus kurang dari ambang batas Anda. Fitur ini dimatikan secara default. Ketika ambang batas ringan ditentukan, pekerjaan profil Anda mengembalikan satu hitungan lagi,`zScoreMildOutliersCount`. Juga, `zScoreOutliersSample` dapat mencakup sampel outlier ambang batas ringan dalam kasus ini.

Pengaturan

- `threshold`— Nilai ambang yang digunakan saat mendeteksi outlier. Nilai ini harus lebih besar atau sama dengan 0.

- `mildThreshold`— Nilai ambang batas ringan untuk digunakan saat mendeteksi outlier. Nilai ini harus lebih besar atau sama dengan 0 dan kurang dari`threshold`.
- `sampleSize`— Ukuran daftar yang mencakup outlier di kolom. Gunakan `"ALL"` untuk daftar semua nilai.

Contoh

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

VALUE_DISTRIBUTION

Mengukur distribusi nilai di kolom dengan rentang nilai. Pekerjaan profil mengelompokkan nilai dari kolom numerik atau kolom string (berdasarkan panjang string) ke dalam bin berdasarkan rentang numerik, dan menghasilkan daftar bin. Tempat sampah berurutan, dan batas atas untuk ember adalah batas bawah untuk ember berikutnya.

Pengaturan

`binNumber`— Jumlah tempat sampah. Nilai ini harus lebih besar dari 0.

Contoh

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

Z_SCORE_DISTRIBUTION

Mengukur distribusi nilai z-skor di kolom numerik. Pekerjaan profil mengelompokkan z-skor nilai ke dalam bin berdasarkan rentang numerik, dan menghasilkan daftar tempat sampah. Tempat sampah berurutan, dan batas atas untuk ember adalah batas bawah untuk ember berikutnya.

Pengaturan

`binNumber`— Jumlah tempat sampah. Nilai ini harus lebih besar dari 0.

Contoh

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

EntityDetectorConfiguration bagian untuk mengkonfigurasi PII

Di `EntityDetectorConfiguration` bagian struktur Anda, Anda dapat mengonfigurasi jenis entitas dalam kumpulan data yang DataBrew ingin Anda deteksi sebagai informasi identifikasi pribadi (PII) untuk pekerjaan profil.

EntityTypes

Anda mengonfigurasi jenis entitas yang DataBrew ingin Anda deteksi sebagai PII untuk pekerjaan profil Anda. Kapan tidak `EntityDetectorConfiguration` terdefinisi, deteksi entitas dinonaktifkan. Jenis entitas berikut dapat dideteksi dalam kumpulan data Anda:

- USA_SSN
- Email
- USA_ITIN
- USA_PASSPORT_NUMBER
- TELEPON_NOMOR
- USA_DRIVING_LICENSE

- BANK_ACCOUNT
- KARTU KREDIT
- IP_ALAMAT
- ALAMAT MAC_
- 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

Grup tipe entitas juga `USA_ALL` didukung, dan mencakup semua jenis entitas di atas kecuali `PERSON_NAME` dan`DATE`.

Tipe `EntityTypes` adalah array string.

AllowedStatistics

Konfigurasikan statistik yang diizinkan untuk dijalankan pada kolom yang berisi entitas yang terdeteksi. Jika tidak `AllowedStatistics` terdefinisi, tidak ada statistik yang akan dihitung pada kolom yang berisi entitas yang terdeteksi. Lihat Statistik yang dapat dikonfigurasi di tingkat kolom daftar nilai yang valid untuk `AllowedStatistics` parameter.

Tipe `AllowedStatistics` adalah array `AllowedStatistics` objek.

Keamanan di AWS Glue DataBrew

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud —AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud.AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Third-partyauditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program Kepatuhan Program AWS Kepatuhan Program AWS . Untuk mempelajari tentang program kepatuhan yang berlaku AWS Glue DataBrew, lihat AWS layanan dalam Layanan Cakupan oleh Program AWS Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Glue DataBrew. Topik berikut menunjukkan cara mengonfigurasi DataBrew untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan DataBrew sumber daya Anda.

Topik

- Perlindungan data di AWS Glue DataBrew
- Identitas dan manajemen akses untuk AWS Glue DataBrew
- Penebangan dan pemantauan di DataBrew
- Validasi kepatuhan untuk AWS Glue DataBrew
- Ketahanan di AWS Glue DataBrew
- Keamanan infrastruktur di AWS Glue DataBrew
- Analisis konfigurasi dan kerentanan di AWS Glue DataBrew

Perlindungan data di AWS Glue DataBrew

DataBrew menawarkan beberapa fitur yang dirancang untuk membantu melindungi data Anda.

Topik

- Enkripsi saat diam
- Enkripsi saat bergerak
- Manajemen kunci
- Mengidentifikasi dan menangani informasi identitas pribadi (PII)
- DataBrew ketergantungan pada yang lain AWS layanan

Model tanggung jawab bersamaAWS berlaku untuk perlindungan data di AWS Glue DataBrew. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat FAQ Privasi Data AWS . Untuk informasi tentang perlindungan data di Eropa, lihat Pusat Peraturan Umum Perlindungan Data (GDPR).

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management(IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.

- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan DataBrew atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi saat diam

DataBrew mendukung enkripsi data saat istirahat untuk DataBrew proyek dan pekerjaan. Proyek dan pekerjaan dapat membaca data terenkripsi, dan pekerjaan dapat menulis data terenkripsi dengan memanggil AWS Key Management Service(AWS KMS) untuk menghasilkan kunci dan mendekripsi data. Anda juga dapat menggunakan kunci KMS untuk mengenkripsi log pekerjaan yang dihasilkan oleh DataBrew pekerjaan. Anda dapat menentukan kunci enkripsi menggunakan DataBrew konsol atau DataBrew API.

> **Important**
>
> AWS Glue DataBrew hanya mendukung tombol AWS KMS simetris. Untuk informasi selengkapnya, lihat kunci AWS KMS di Panduan AWS Key Management Service Pengembang.

Saat Anda membuat pekerjaan DataBrew dengan enkripsi diaktifkan, Anda dapat menggunakan DataBrew konsol untuk menentukan kunci enkripsi S3-managed sisi server (SSE-S3) atau kunci KMS yang disimpan di AWS KMS(SSE-KMS) untuk mengenkripsi data saat istirahat.

> **Important**
>
> Saat Anda menggunakan kumpulan data Amazon Redshift, objek yang diturunkan ke direktori sementara yang disediakan akan dienkripsi. SSE-S3

Mengenkripsi data yang ditulis oleh pekerjaan DataBrew

DataBrew pekerjaan dapat menulis ke target Amazon S3 terenkripsi dan Log Amazon terenkripsi. CloudWatch

Topik

- Menyiapkan DataBrew untuk menggunakan enkripsi
- Membuat rute ke AWS KMS untuk pekerjaan VPC
- Menyiapkan enkripsi dengan AWS Kunci KMS

Menyiapkan DataBrew untuk menggunakan enkripsi

Ikuti prosedur ini untuk mengatur DataBrew lingkungan Anda agar menggunakan enkripsi.

Untuk mengatur DataBrew lingkungan Anda untuk menggunakan enkripsi

1. Buat atau perbarui kunci AWS KMS Anda untuk memberikan AWS KMS izin ke peran AWS Identity and Access Management(IAM) yang diteruskan ke pekerjaan. DataBrew Peran IAM ini digunakan untuk mengenkripsi CloudWatch Log dan target Amazon S3. Untuk informasi selengkapnya, lihat Mengenkripsi Data Log di CloudWatch Log Menggunakan AWS KMS di Panduan Pengguna Amazon CloudWatch Logs.

   Dalam contoh berikut,, *"role1""role2"*, dan *"role3"* merupakan peran IAM yang diteruskan ke DataBrew pekerjaan. Pernyataan kebijakan ini menjelaskan kebijakan kunci KMS yang memberikan izin ke peran IAM yang terdaftar untuk mengenkripsi dan mendekripsi dengan kunci KMS ini.

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

```
            "kms:Decrypt*",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:Describe*"
        ],
        "Resource": "*"
    }
```

`Service`Pernyataan, ditampilkan sebagai`"Service": "logs.region.amazonaws.com"`, diperlukan jika Anda menggunakan kunci untuk mengenkripsi CloudWatch Log.

2. Pastikan AWS KMS kunci diatur `ENABLED` sebelum digunakan.

Untuk informasi selengkapnya tentang menentukan izin menggunakan kebijakan AWS KMS utama, lihat Menggunakan kebijakan utama di.AWS KMS

Membuat rute ke AWS KMS untuk pekerjaan VPC

Anda dapat connect langsung ke AWS KMS melalui titik akhir privat di virtual private cloud (VPC) Anda alih-alih terhubung melalui internet. Saat Anda menggunakan titik akhir VPC, komunikasi antara VPC Anda dan AWS KMS dilakukan sepenuhnya di dalam jaringan.AWS

Anda dapat membuat titik akhir AWS KMS VPC dalam VPC. Tanpa langkah ini, DataBrew pekerjaan Anda mungkin gagal dengan a`kms timeout`. Untuk petunjuk terperinci, lihat Menghubungkan ke AWS KMS Melalui Titik Akhir VPC di Panduan Pengembang.AWS Key Management Service

Saat Anda mengikuti petunjuk ini, pada konsol VPC, pastikan untuk melakukan hal berikut:

- Pilih Aktifkan nama DNS Pribadi.
- Untuk grup Keamanan, pilih grup keamanan (termasuk aturan referensi mandiri) yang Anda gunakan untuk DataBrew pekerjaan yang mengakses Java Database Connectivity (JDBC).

Ketika Anda menjalankan DataBrew pekerjaan yang mengakses penyimpanan data JDBC, DataBrew harus memiliki rute ke titik akhir.AWS KMS Anda dapat memberikan rute dengan gateway terjemahan alamat jaringan (NAT) atau dengan titik akhir AWS KMS VPC. Untuk membuat gateway NAT, lihat Gateway NAT di Panduan Pengguna Amazon VPC.

Menyiapkan enkripsi dengan AWS Kunci KMS

Saat Anda mengaktifkan enkripsi pada suatu pekerjaan, itu berlaku untuk Amazon S3 dan. CloudWatch Peran IAM yang diteruskan harus memiliki AWS KMS izin berikut.

Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon:

- Untuk selengkapnya`SSE-S3`, lihat Melindungi Data Menggunakan Server-Side Enkripsi dengan Kunci S3-Managed Enkripsi Amazon (SSE-S3).
- Untuk selengkapnya`SSE-KMS`, lihat Melindungi Data Menggunakan Server-Side Enkripsi dengan AWS KMS—Kunci Terkelola (). SSE-KMS

Enkripsi saat bergerak

AWS menyediakan enkripsi Secure Sockets Layer (SSL) untuk data dalam penerbangan.

DataBrew dukungan untuk sumber data JDBC datang melalui.AWS Glue Saat menghubungkan ke sumber data JDBC, DataBrew gunakan pengaturan pada AWS Glue koneksi Anda, termasuk opsi Memerlukan koneksi SSL. Untuk informasi selengkapnya, lihat Properti AWS Glue Koneksi -AWS Glue di Panduan AWS Glue Pengembang.

AWS KMS menyediakan enkripsi “bawa kunci Anda sendiri” dan enkripsi sisi server untuk pemrosesan DataBrew ekstrak, transformasi, muat (ETL) dan untuk.AWS Glue Data Catalog

Manajemen kunci

Anda dapat menggunakan IAM DataBrew untuk menentukan pengguna,AWS sumber daya, grup, peran, dan kebijakan berbutir halus mengenai akses, penolakan, dan lainnya.

Anda dapat menentukan akses ke metadata menggunakan kebijakan berbasis sumber daya dan berbasis identitas, tergantung pada kebutuhan organisasi Anda. Resource-based kebijakan mencantumkan prinsipal yang diizinkan atau ditolak akses ke sumber daya Anda, memungkinkan Anda menyiapkan kebijakan seperti akses lintas akun. Kebijakan identitas secara khusus dilampirkan pada pengguna, grup, dan peran dalam IAM.

DataBrew mendukung pembuatan enkripsi AWS KMS key“bawa kunci Anda sendiri” Anda sendiri. DataBrew juga menyediakan enkripsi sisi server menggunakan kunci KMS dari untuk pekerjaan.AWS KMS DataBrew

Mengidentifikasi dan menangani informasi identitas pribadi (PII)

Saat Anda membangun fungsi analitik atau model pembelajaran mesin, Anda memerlukan perlindungan untuk mencegah paparan data informasi identifikasi pribadi (PII). PII adalah data pribadi yang dapat digunakan untuk mengidentifikasi seseorang, seperti alamat, nomor rekening bank, atau nomor telepon. Misalnya, ketika analis data dan ilmuwan data menggunakan kumpulan data untuk menemukan informasi demografis umum, mereka seharusnya tidak memiliki akses ke PII individu tertentu.

DataBrew menyediakan mekanisme penyembunyian data untuk mengaburkan data PII selama proses persiapan data. Bergantung pada kebutuhan organisasi Anda, ada berbagai mekanisme redaksi data PII yang tersedia. Anda dapat mengaburkan data PII sehingga pengguna tidak dapat mengembalikannya kembali, atau Anda dapat membuat obfuscation reversibel.

Mengidentifikasi dan menutupi data PII DataBrew melibatkan pembuatan serangkaian transformasi yang dapat digunakan pelanggan untuk menyunting data PII. Bagian dari proses ini adalah menyediakan deteksi dan statistik data PII di dasbor ikhtisar Profil Data di DataBrew konsol.

Anda dapat menggunakan teknik masking data berikut:

- Substitusi - Ganti data PII dengan nilai yang tampak otentik lainnya.
- Shuffling - Kocokkan nilai dari kolom yang sama di baris yang berbeda.
- Enkripsi deterministik — Menerapkan algoritma enkripsi deterministik ke nilai kolom. Enkripsi deterministik selalu menghasilkan ciphertext yang sama untuk suatu nilai.
- Enkripsi probabilistik — Menerapkan algoritma enkripsi probabilistik ke nilai kolom. Enkripsi probabilistik menghasilkan ciphertext yang berbeda setiap kali diterapkan.
- Dekripsi — Dekripsi kolom berdasarkan kunci enkripsi.
- Nulling out atau penghapusan - Ganti bidang tertentu dengan nilai null atau hapus kolom.
- Masking out — Gunakan karakter scrambling atau menutupi bagian-bagian tertentu dalam kolom.
- Hashing - Terapkan fungsi hash ke nilai kolom.

Untuk informasi selengkapnya tentang penggunaan transformasi, lihat Langkah resep Informasi Identifikasi Pribadi (PII). Untuk informasi selengkapnya tentang penggunaan pekerjaan profil untuk mendeteksi PII, termasuk daftar tipe entitas yang dapat dideteksi, lihat EntityDetectorConfiguration bagian untuk mengonfigurasi PII di Membuat konfigurasi pekerjaan profil secara terprogram.

DataBrew ketergantungan pada yang lain AWS layanan

Untuk bekerja dengan DataBrew konsol, Anda memerlukan serangkaian izin minimum untuk bekerja dengan DataBrew sumber daya untuk AWS akun Anda. Selain DataBrew izin ini, konsol memerlukan izin dari layanan berikut:

- CloudWatch Log izin untuk menampilkan log.
- Izin IAM untuk membuat daftar dan meneruskan peran.
- Izin Amazon EC2 untuk mencantumkan VPC, subnet, grup keamanan, instans, dan objek lainnya. DataBrew menggunakan izin ini untuk menyiapkan item Amazon EC2 seperti VPC saat menjalankan pekerjaan. DataBrew
- Izin Amazon S3 untuk mencantumkan bucket dan objek.
- AWS Glue izin untuk membaca objek AWS Glue skema, seperti database, partisi, tabel, dan koneksi.
- AWS Lake Formation izin untuk bekerja dengan danau data Lake Formation.

Identitas dan manajemen akses untuk AWS Glue DataBrew

AWS Identity and Access Management(IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. DataBrew IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- AWS Glue DataBrew and AWS Lake Formation
- Bagaimana AWS Glue DataBrew bekerja dengan IAM
- Identity-based contoh kebijakan untuk AWS Glue DataBrew
- AWS kebijakan terkelola untuk AWS Glue DataBrew
- Memecahkan masalah identitas dan akses di AWS Glue DataBrew

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensional dari sumber identitas seperti AWS IAM Identity Center(Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensional. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In.

Untuk akses terprogram,AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Pengguna dan grup

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan

memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya.AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Identity-based kebijakan

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Identity-based kebijakan dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada beberapa identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Resource-based kebijakan

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3.

Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

Resource-based kebijakan adalah kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

DataBrew tidak mendukung kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3,AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat Gambaran umum daftar kontrol akses (ACL) dalam Panduan Developer Amazon Simple Storage Service.

DataBrew tidak mendukung ACL.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – Menentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations.
- Kebijakan kontrol sumber daya (RCP) – Menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCP) dalam Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

AWS Glue DataBrew and AWS Lake Formation

AWS Glue DataBrew mendukung AWS Lake Formation izin untuk AWS Glue Data Catalog tabel. Ketika kumpulan data menggunakan AWS Glue Data Catalog tabel yang terdaftar dengan Lake Formation, peran IAM yang diberikan untuk proyek atau pekerjaan harus memiliki izin DESCRIPTION dan SELECT Lake Formation di atas tabel.

AWS Glue DataBrew mendukung penulisan ke AWS Glue Data Catalog tabel berdasarkan AWS Lake Formation. Ketika DataBrew pekerjaan menggunakan Katalog Data yang terdaftar di Lake Formation, peran IAM yang diberikan untuk pekerjaan harus memiliki izin INSERT, ALTER, dan DELETE dari Lake Formation untuk tabel yang terlibat. Peran IAM harus memiliki `glue:UpdateTable` izin, dan juga izin ke lokasi data yang terkait dengan tabel Katalog Data.

Bagaimana AWS Glue DataBrew bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses DataBrew, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan. DataBrew Untuk mendapatkan tampilan tingkat tinggi tentang cara DataBrew dan AWS layanan lain bekerja dengan IAM, lihat AWS Layanan yang Bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- DataBrew kebijakan berbasis identitas
- Resource-based kebijakan di DataBrew
- DataBrew Peran IAM

DataBrew kebijakan berbasis identitas

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, dan juga ketentuan di mana tindakan tersebut diperbolehkan atau ditolak. DataBrew mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat Referensi Elemen Kebijakan JSON IAM dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, kebijakan AWS JSON dapat menentukan prinsipal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen Tindakan kebijakan JSON menjelaskan tindakan yang dapat Anda izinkan atau tolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama sebagaimana operasi API AWS yang dikaitkan padanya. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan DataBrew menggunakan awalan berikut sebelum tindakan:`databrew:`. Misalnya, untuk memberikan izin kepada seseorang untuk menjalankan instans Amazon EC2 dengan operasi API `RunInstances` Amazon EC2, Anda menyertakan tindakan `ec2:RunInstances` dalam kebijakan mereka. Pernyataan kebijakan harus mencakup salah satu `Action` atau `NotAction` elemen. DataBrew mendefinisikan serangkaian tindakannya sendiri yang menggambarkan tugas yang dapat Anda lakukan dengannya.

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan tindakan dengan koma seperti berikut:

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

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

Untuk melihat daftar tindakan, lihat DataBrew Tindakan yang Ditentukan oleh AWS Glue DataBrew dalam Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Berikut ini adalah DataBrew API yang tidak mendukung izin tingkat sumber daya:

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

Sumber daya DataBrew kumpulan data memiliki Nama Sumber Daya Amazon (ARN) berikut.

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

Untuk informasi selengkapnya tentang format ARN, lihat Nama Sumber Daya Amazon (ARN) dan Ruang Nama AWS Layanan.

Misalnya, untuk menentukan `i-1234567890abcdef0` instance dalam pernyataan Anda, gunakan ARN berikut.

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

Untuk menentukan semua instance milik akun tertentu, gunakan wildcard (*).

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

Anda tidak dapat melakukan beberapa DataBrew tindakan, seperti untuk membuat sumber daya, pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

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

Untuk melihat daftar jenis DataBrew sumber daya dan ARNnya, lihat Sumber Daya yang Ditentukan oleh AWS Glue DataBrew dalam Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat menentukan ARN setiap sumber daya, lihat Tindakan yang Ditentukan oleh AWS Glue DataBrew.

Kunci syarat

DataBrew tidak menyediakan kunci kondisi khusus layanan apa pun, tetapi mendukung penggunaan beberapa kunci kondisi global. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Contoh

Untuk melihat contoh kebijakan DataBrew berbasis identitas, lihat. Identity-based contoh kebijakan untuk AWS Glue DataBrew

Resource-based kebijakan di DataBrew

DataBrew tidak mendukung kebijakan berbasis sumber daya.

DataBrew Peran IAM

Peran IAM adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensyal sementara dengan DataBrew

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda mendapatkan kredensil keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

DataBrew mendukung menggunakan kredensil sementara.

Service-linked peran

Service-linked peran memungkinkan AWS layanan mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Service-linked peran muncul di akun IAM Anda dan dimiliki oleh layanan. Administrator dapat melihat tetapi tidak dapat mengedit izin untuk peran yang terkait dengan layanan.

Memilih peran IAM di DataBrew

Saat membuat sumber daya kumpulan data DataBrew, Anda memilih peran IAM untuk mengizinkan DataBrew akses atas nama Anda. Jika sebelumnya Anda telah membuat peran layanan atau peran

terkait layanan, DataBrew berikan daftar peran yang dapat dipilih. Pastikan untuk memilih peran yang memungkinkan akses baca ke bucket atau AWS Glue Data Catalog sumber daya Amazon S3, yang sesuai.

Identity-based contoh kebijakan untuk AWS Glue DataBrew

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya DataBrew . Mereka juga tidak dapat melakukan tugas menggunakan Konsol Manajemen AWS,AWS CLI, atau AWS API. Administrator harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya tertentu yang mereka butuhkan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat Kebijakan pada Tab JSON dalam Panduan Pengguna IAM.

Topik

- Praktik terbaik kebijakan
- Menggunakan DataBrew konsol
- Memungkinkan pengguna untuk melihat izin mereka sendiri
- Mengelola DataBrew sumber daya berdasarkan tag

Praktik terbaik kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus DataBrew sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda,Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan DataBrew konsol

Untuk mengakses AWS Glue DataBrew konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang DataBrew sumber daya di AWS akun Anda. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak berfungsi sebagaimana dimaksudkan untuk pengguna atau peran dengan kebijakan tersebut.

Untuk memastikan bahwa pengguna dan peran dapat menggunakan DataBrew konsol, lampirkan juga kebijakan AWS terkelola berikut ke entitas. Untuk informasi selengkapnya, lihat Menambahkan Izin ke Pengguna dalam Panduan Pengguna IAM.

```
AWSDataBrewConsoleAccess
```

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau DataBrew API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Memungkinkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram.AWS CLIAWS

```
{
    "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": "*"
        }
    ]
}
```

Mengelola DataBrew sumber daya berdasarkan tag

Anda dapat menggunakan kondisi dalam kebijakan berbasis identitas untuk mengelola DataBrew sumber daya berdasarkan tag, misalnya, untuk menghapus, memperbarui, atau menjelaskan sumber daya. Contoh berikut menunjukkan kebijakan yang menyangkal penghapusan proyek. Namun, penghapusan ditolak hanya jika pemilik tag proyek memiliki nilai admin. Kebijakan ini juga memberikan izin yang diperlukan untuk menolak tindakan ini di konsol.

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"}
         }
      }
   ]
}
```

Anda dapat melampirkan kebijakan ini ke pengguna di akun Anda. Jika pengguna bernama richard-roe mencoba menghapus DataBrew proyek, sumber daya tidak boleh diberi tag owner=admin atau owner=admin. Jika tidak, pengguna ditolak izin untuk menghapus proyek. Kunci tag kondisi Pemilik cocok dengan Pemilik dan pemilik karena nama kunci kondisi tidak peka huruf besar/kecil. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM JSON: Syarat dalam Panduan Pengguna IAM.

> **Note**
>
> ListDatasets, ListJobs, ListProjects, ListRecipes, ListRulesets, dan ListSchedules tidak mendukung kontrol akses berbasis tag.

AWS kebijakan terkelola untuk AWS Glue DataBrew

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan yang dikelola pelanggan IAM yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di AWS akun Anda. Untuk informasi selengkapnya tentang kebijakan AWSAWS terkelola, lihat kebijakan terkelola di Panduan Pengguna IAM.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan AWS terkelola untuk mendukung fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau saat operasi baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu,AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan ReadOnlyAccessAWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru,AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk daftar dan deskripsi kebijakan fungsi pekerjaan, lihat kebijakan AWS terkelola untuk fungsi pekerjaan di Panduan Pengguna IAM.

DataBrew update ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola DataBrew sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan

RSS di halaman Riwayat DataBrew dokumen. Kebijakan terkelola dapat ditemukan di konsol AWS IAM di AwsGlueDataBrewFullAccessPolicy.

Ubah	Deskripsi	Date
AWSGlueDataBrewServiceRole— Izin baca untuk AWS Glue ditambahkan.	Pembaruan ini menambahkan `glue:GetCustomEntityType` . Izin ini diperlukan untuk menjalankan pekerjaan AWS Glue DataBrew profil dengan PII-identification diaktifkan.	Maret 20, 2024
AWSGlueDataBrewServiceRole- Baca izin untuk AWS Glue ditambahkan.	Pembaruan ini menambahkan `glue:BatchGetCustomEntityTypes` . Izin ini diperlukan untuk menjalankan pekerjaan AWS Glue DataBrew profil dengan PII-identification diaktifkan.	9 Mei 2022
AwsGlueDataBrewFullAccessPolicy- Baca izin untuk Amazon Redshift-Data DescribeStatements dan Amazon GetLifecycleConfiguration S3 ditambahkan.	Pembaruan ini menambah dukungan `redshift-data:DescribeStatement` untuk memvalidasi SQL Anda saat membuat kumpulan data Amazon Redshift-based . Ini juga menambah `s3:GetLifecycleConfiguration` untuk mengevaluasi apakah awalan bucket Amazon S3 yang Anda berikan sebagai direktori sementara memiliki siklus hidup yang dikonfigurasi atau tidak. Selain itu, perubahan ini menggantikan	4 Februari 2022

Ubah	Deskripsi	Date
	izin “databrew: *” dengan daftar izin eksplisit termasuk semua API. DataBrew	
AwsGlueDataBrewFullAccessPolicy- Read/write izin untuk AWS Secrets Manager ditambahkan.	Pembaruan ini menambahkan `secretsmanager:CreateSecret` dan `secretsmanager:GetSecretValue` untuk rahasia bernama `databrew!default` , rahasia default untuk digunakan dengan DataBrew transformasi. Selain itu, ia menambahkan izin CreateSecret untuk rahasia yang diawali dengan `AwsGlueDataBrew-` untuk membuat rahasia dari konsol. DataBrew GenerateRandom, dijelaskan dalam Referensi AWS Key Management Service API, digunakan untuk menghasilkan string byte acak yang aman secara kriptografis.	18 November 2021
AWSGlueDataBrewServiceRole- Read/write izin untuk AWS Secrets Manager ditambahkan.	Pembaruan ini menambahkan `secretsmanager:GetSecretValue` rahasia bernama `databrew!default` , rahasia default untuk digunakan dengan DataBrew transformasi.	18 November 2021

Ubah	Deskripsi	Date
AwsGlueDataBrewFullAccessPolicy- Read/write izin untuk AWS Secrets Manager ditambahkan.	Pembaruan ini menambahkan `secretsmanager:CreateSecret` dan `secretsmanager:GetSecretValue` untuk rahasia bernama`databrew!default`, rahasia default untuk digunakan dengan DataBrew transformasi. Selain itu, ia menambahkan izin CreateSecret untuk rahasia yang diawali dengan `AwsGlueDataBrew-` untuk membuat rahasia dari konsol. DataBrew `kms:GenerateRandom` (`https://docs.aws.amazon.com/kms/latest/APIReference/API_GenerateRandom.html`) digunakan untuk menghasilkan string byte acak yang aman secara kriptografis.	18 November 2021
AWSGlueDataBrewServiceRole- Read/write izin untuk AWS Secrets Manager ditambahkan.	Pembaruan ini menambahkan `secretsmanager:GetSecretValue` rahasia bernama`databrew!default`, rahasia default untuk digunakan dengan DataBrew transformasi.	18 November 2021

Ubah	Deskripsi	Date
AwsGlueDataBrewFullAccessPolicy- Baca izin untuk database AWS Glue katalog dan buat izin untuk tabel AWS Glue katalog ditambahkan.	Pembaruan ini menambahkan izin untuk mencantumkan database AWS Glue Katalog dan membuat tabel katalog baru di bawah database yang ada sebagai bagian dari konfigurasi output ke pekerjaan. DataBrew	30 Juni 2021
AwsGlueDataBrewFullAccessPolicy- Read/write izin untuk fitur AppFlow dataset Amazon ditambahkan.	Pembaruan ini menambahkan izin untuk membaca AppFlow alur Amazon dan eksekusi aliran yang ada dan untuk membuat eksekusi aliran.	28 April 2021
AwsGlueDataBrewFullAccessPolicy- Baca izin untuk dataset database ditambahkan.	Pembaruan ini menambahkan izin untuk membaca AWS Glue koneksi yang ada dan membuat AWS Glue koneksi baru untuk digunakan. DataBrew Selain itu, untuk membuat pengalaman konsol dalam membuat koneksi baru lebih mudah, ini memungkinkan daftar sumber daya VPC Amazon dan cluster Amazon Redshift. Ini juga memberikan izin untuk membuat daftar, tetapi tidak membaca,AWS Secrets Manager rahasia.	30 Maret 2021

Ubah	Deskripsi	Date
DataBrew mulai melacak perubahan	DataBrew mulai melacak perubahan untuk kebijakan yang AWS dikelola.	30 Maret 2021

Memecahkan masalah identitas dan akses di AWS Glue DataBrew

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan DataBrew dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di DataBrew
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya AWS akun untuk mengakses DataBrew sumber daya saya

Saya tidak berwenang untuk melakukan tindakan di DataBrew

Jika Konsol Manajemen AWS memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, hubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi ketika `mateojackson` pengguna mencoba menggunakan konsol untuk melihat detail tentang proyek tetapi tidak memiliki `databrew:DescribeProject` izin.

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

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk mengizinkan dia mengakses sumber daya *`my-example-project`* menggunakan tindakan `databrew:`*`GetProject`*.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran DataBrew.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di DataBrew. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

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

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya AWS akun untuk mengakses DataBrew sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah DataBrew mendukung fitur-fitur ini, lihatBagaimana AWS Glue DataBrew bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.

- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Penebangan dan pemantauan di DataBrew

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja DataBrew dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multipoint jika terjadi.AWS menyediakan beberapa alat untuk memantau DataBrew sumber daya Anda dan menanggapi potensi insiden:

CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm Amazon, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan akan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon SNS. CloudWatch alarm tidak memanggil tindakan karena mereka berada dalam keadaan tertentu. Sebaliknya, negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu.

AWS CloudTrail Log

CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di DataBrew. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat DataBrew, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Validasi kepatuhan untuk AWS Glue DataBrew

Third-party auditor menilai keamanan dan kepatuhan AWS Glue DataBrew sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat Dokumentasi AWS Keamanan.

Ketahanan di AWS Glue DataBrew

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan.AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk AWS Glue DataBrew, kami menyarankan Anda mengonfigurasi pekerjaan Anda untuk menggunakan satu atau beberapa percobaan ulang. Jumlah percobaan ulang untuk pekerjaan dikonfigurasi di DataBrew konsol di bawah Pengaturan pekerjaan lanjutan.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan infrastruktur di AWS Glue DataBrew

Sebagai bagian dari layanan terkelola,AWS Glue DataBrew dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper Amazon Web Services: Tinjauan Proses Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses DataBrew melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.0 atau versi yang lebih baru. Kami merekomendasikan TLS 1.2 atau versi yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS) seperti Ephemeral (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan principal IAM. Atau Anda dapat menggunakan AWS Security Token

Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Topik

- Penggunaan AWS Glue DataBrew dengan VPC Anda
- Penggunaan AWS Glue DataBrew dengan titik akhir VPC

Penggunaan AWS Glue DataBrew dengan VPC Anda

Jika Anda menggunakan Amazon VPC untuk meng-host AWS sumber daya Anda, Anda dapat mengonfigurasi AWS Glue DataBrew untuk merutekan lalu lintas melalui virtual private cloud (VPC) berdasarkan layanan Amazon VPC. DataBrew melakukan ini dengan terlebih dahulu menyediakan elastic network interface di subnet yang Anda tentukan. DataBrew kemudian melampirkan grup keamanan yang Anda tentukan ke antarmuka jaringan tersebut untuk mengontrol akses. Grup keamanan yang ditentukan harus memiliki aturan masuk dan keluar referensi sendiri untuk semua lalu lintas. Selain itu, VPC Anda harus mengaktifkan nama host dan resolusi DNS. Untuk informasi selengkapnya, lihat Menyiapkan VPC agar Terhubung ke Toko Data JDBC di Panduan Pengembang.AWS Glue

Untuk AWS Glue Data Catalog kumpulan data, informasi VPC dikonfigurasi saat Anda membuat AWS Glue sambungan di Katalog Data. Untuk membuat tabel Katalog Data untuk koneksi ini, jalankan crawler dari AWS Glue konsol. Untuk informasi selengkapnya, lihat Mengisi AWS Glue Data Catalog di Panduan AWS Glue Pengembang.

Untuk kumpulan data database, tentukan informasi VPC Anda saat Anda membuat koneksi dari konsol. DataBrew

Untuk menggunakan AWS Glue DataBrew subnet VPC tanpa NAT, Anda harus memiliki titik akhir VPC gateway ke Amazon S3 dan titik akhir VPC untuk antarmuka.AWS Glue Untuk informasi selengkapnya, lihat Membuat titik akhir gateway dan titik akhir VPC Antarmuka AWS PrivateLink() di dokumentasi VPC Amazon. Antarmuka elastis yang disediakan oleh DataBrew tidak memiliki alamat IPv4 publik, sehingga tidak mendukung penggunaan Internet Gateway VPC.

Titik akhir antarmuka Amazon S3 tidak didukung saat ini. Jika Anda menggunakan AWS Secrets Manager untuk menyimpan rahasia Anda, Anda memerlukan rute ke Secrets Manager. Jika Anda menggunakan enkripsi, Anda memerlukan rute ke AWS Key Management Service(AWS KMS).

Penggunaan AWS Glue DataBrew dengan titik akhir VPC

Jika Anda menggunakan Amazon VPC untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC Anda dan dengan DataBrew menyediakan titik akhir VPC. Dengan menggunakan titik akhir VPC ini, Anda dapat melakukan DataBrew panggilan API.

Endpoint DataBrew VPC tidak diperlukan untuk digunakan dengan VPC DataBrew Anda. Untuk informasi selengkapnya, lihat Penggunaan AWS Glue DataBrew dengan VPC Anda.

Anda dapat menggunakan AWS Glue dengan titik akhir VPC di semua AWS Wilayah yang mendukung keduanya dan titik akhir AWS Glue VPC.

Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna Amazon VPC:

- Apa yang Dimaksud Amazon VPC?
- Membuat Endpoint Antarmuka

Analisis konfigurasi dan kerentanan di AWS Glue DataBrew

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat model tanggung jawab AWS bersama.

Memantau AWS Glue DataBrew

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja AWS Glue DataBrew dan AWS solusi Anda yang lain.AWS menyediakan alat pemantauan berikut untuk menonton DataBrew, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Events memungkinkan Anda mengatur notifikasi otomatis untuk acara tertentu di DataBrew. Acara dari DataBrew dikirim ke CloudWatch Acara dalam waktu nyaris nyata. Anda dapat mengonfigurasi CloudWatch Acara untuk memantau peristiwa dan memanggil target sebagai respons terhadap peristiwa yang menunjukkan perubahan pada pembagian sumber daya Anda. Perubahan pada pembagian sumber daya memicu peristiwa untuk pemilik pembagian sumber daya dan prinsipal yang diberikan akses ke pembagian sumber daya. Untuk informasi selengkapnya, lihat Panduan Pengguna CloudWatch Acara Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans Amazon EC2, CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda. Kemudian, mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudTrail.

Topik

- Pemantauan DataBrew dengan Amazon CloudWatch
- Mengotomatisasi DataBrew dengan Acara CloudWatch

- Pemantauan DataBrew dengan CloudWatch Log
- Logging panggilan DataBrew API dengan AWS CloudTrail
- Penggunaan AWS Pemberitahuan Pengguna dengan AWS Glue Databrew

Pemantauan DataBrew dengan Amazon CloudWatch

Anda dapat memantau DataBrew penggunaan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

AWS Glue DataBrew melaporkan metrik berikut di `AWS/DataBrew` namespace.

Metrik	Deskripsi
`SessionCount`	Jumlah total DataBrew sesi di seluruh akun pelanggan Dimensi yang Valid: LogGroupName Statistik Valid: Sum Unit: Hitungan

Mengotomatisasi DataBrew dengan Acara CloudWatch

Amazon CloudWatch Events memungkinkan Anda mengotomatiskan AWS layanan dan merespons secara otomatis peristiwa sistem seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirimkan ke CloudWatch Acara dalam waktu nyaris nyata. Anda dapat menulis aturan sederhana untuk menunjukkan kejadian mana yang sesuai kepentingan Anda, dan tindakan otomatis apa yang diambil ketika suatu kejadian sesuai dengan suatu aturan. Tindakan yang dapat dipicu secara otomatis meliputi hal-hal berikut:

- Memanggil perintah Amazon EC2 run
- Mengirim peristiwa ke Amazon Kinesis Data Streams

- Mengaktifkan mesin AWS Step Functions negara
- Memberi tahu topik Amazon SNS atau antrean Amazon SQS

DataBrew melaporkan peristiwa ke CloudWatch Acara setiap kali status sumber daya di AWS akun Anda berubah. Peristiwa dipancarkan atas dasar upaya terbaik.

Berikut ini adalah contoh dari beberapa peristiwa, menunjukkan berbagai keadaan DataBrew pekerjaan:`SUCCEEDED`,`FAILED`,`TIMEOUT`, dan`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"
  }
}
```

Untuk informasi selengkapnya, lihat Panduan Pengguna CloudWatch Acara Amazon.

Pemantauan DataBrew dengan CloudWatch Log

Anda dapat memantau DataBrew pekerjaan menggunakan CloudWatch Log, yang mengumpulkan informasi terperinci dari subsistem DataBrew pekerjaan dan membuatnya tersedia untuk ditinjau. Log ini dapat membantu jika Anda ingin mendapatkan wawasan tentang sumber daya yang digunakan profil dan pekerjaan resep Anda, atau untuk tujuan pemecahan masalah, Untuk informasi selengkapnya, lihat Panduan Pengguna CloudWatch Log Amazon.

Logging panggilan DataBrew API dengan AWS CloudTrail

DataBrew terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di DataBrew. CloudTrail menangkap semua panggilan API untuk DataBrew sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari DataBrew konsol dan panggilan kode ke operasi DataBrew API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. DataBrew Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat DataBrew. Anda juga dapat menentukan alamat IP untuk membuat permintaan, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

DataBrew Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di DataBrew, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat CloudTrail Acara dengan Riwayat Acara di Panduan AWS CloudTrail Pengguna.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk DataBrew, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam

CloudTrail log. Untuk informasi selengkapnya, pelajari topik berikut di Panduan Pengguna AWS CloudTrail:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun

Semua DataBrew tindakan dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API. Misalnya, panggilan ke`CreateDataset`, `UpdateRecipe` dan `StartJobRun` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna atau root.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Memahami Entri File DataBrew Log

Sekali lagi, CloudTrail jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateProfileJob` operasi.

```
{
    "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"
}
```

Penggunaan AWS Pemberitahuan Pengguna dengan AWS Glue Databrew

Anda dapat menggunakan Pemberitahuan AWS Pengguna untuk mengatur saluran pengiriman agar mendapat pemberitahuan tentang peristiwa AWS Glue Databrew. Anda akan menerima notifikasi saat ada sebuah peristiwa yang cocok dengan sebuah aturan yang Anda tentukan. Anda dapat menerima pemberitahuan untuk acara melalui beberapa saluran, termasuk email, Pengembang Amazon Q dalam pemberitahuan obrolan aplikasi obrolan, atau pemberitahuan AWS Console Mobile Applicationpush. Anda juga dapat melihat notifikasi di Pusat Notifikasi Konsol.AWS Pemberitahuan

Pengguna mendukung agregasi, yang dapat mengurangi jumlah notifikasi yang Anda terima selama acara tertentu.

Langkah resep dan referensi fungsi

Dalam referensi ini, Anda dapat menemukan deskripsi langkah-langkah resep dan fungsi yang dapat Anda gunakan secara terprogram, baik dari AWS CLI atau dengan menggunakan salah satu SDK.AWS Dalam DataBrew, langkah resep adalah tindakan yang mengubah data mentah Anda menjadi formulir yang siap dikonsumsi oleh pipa data Anda. DataBrew Fungsi adalah jenis langkah resep khusus yang melakukan perhitungan berdasarkan parameter.

Kategori untuk transformasi di UI meliputi:

- Langkah-langkah resep kolom dasar
  - Filter
  - Kolom
- Langkah resep pembersihan data
  - Format
  - Bersih
  - Ekstrak
- Langkah-langkah resep kualitas data
  - Hilang
  - Tidak valid
  - Duplikat
  - Pencilan
- Langkah-langkah resep informasi pribadi (PII)
  - Masker informasi pribadi
  - Ganti informasi pribadi
  - Enkripsi informasi pribadi
  - Baris acak
- Langkah-langkah resep struktur kolom
  - Membagi
  - Gabungkan
  - Buat
- Langkah resep pemformatan kolom

  - Presisi desimal
  - Ribuan pemisah
  - Singkat angka
- Langkah-langkah resep struktur data
  - Nest-Unnest
  - Pivot
  - Kelompok
  - Join
  - Union
- Langkah-langkah resep ilmu data
  - Teks
  - Penskalaan
  - Pemetaan
  - Mengkodekan
- Fungsi
  - Fungsi matematika
  - Fungsi agregat
  - Fungsi teks
  - Fungsi tanggal dan waktu
  - Fungsi jendela
  - Fungsi web
  - Fungsi lainnya

Untuk informasi selengkapnya tentang bagaimana langkah dan fungsi resep ini digunakan dalam resep (termasuk penggunaan ekspresi kondisi) lihatMendefinisikan struktur resep.

Bagian berikut menjelaskan langkah dan fungsi resep, yang diatur oleh apa yang mereka lakukan.

Topik

- Langkah-langkah resep kolom dasar
- Langkah-langkah resep pembersihan data
- Langkah-langkah resep kualitas data

- Langkah-langkah resep informasi identitas pribadi (PII)
- Deteksi outlier dan langkah-langkah penanganan resep
- Langkah-langkah resep struktur kolom
- Langkah resep pemformatan kolom
- Langkah-langkah resep struktur data
- Langkah-langkah resep ilmu data
- Fungsi matematika
- Fungsi agregat
- Fungsi teks
- Fungsi tanggal dan waktu
- Fungsi jendela
- Fungsi web
- Fungsi lainnya

Langkah-langkah resep kolom dasar

Gunakan tindakan resep kolom dasar ini untuk melakukan transformasi sederhana pada data Anda.

Topik

- CHANGE_DATA_TYPE
- DELETE
- MENGGANDAKAN
- JSON_TO_STRUCTS
- BERGERAK_SETELAH
- BERGERAK_SEBELUM
- MOVE_TO_END
- MOVE_TO_INDEX
- MOVE_TO_START
- GANTI NAMA
- SORT
- TO_BOOLEAN_COLUMN

- KE_DOUBLE_COLUMN
- TO_NUMBER_COLUMN
- TO_STRING_COLUMN

CHANGE_DATA_TYPE

Mengubah tipe data dari kolom yang ada.

Jika nilai kolom tidak dapat dikonversi ke tipe baru, itu akan diganti dengan NULL. Hal ini dapat terjadi ketika kolom string dikonversi ke kolom integer. Misalnya, string “123" akan menjadi bilangan bulat 123, tetapi string “ABC” tidak dapat menjadi angka, sehingga akan diganti dengan nilai NULL.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe kolom baru. tipe data berikut didukung:
  - byte: Nomor integer bertanda 1-byte. Kisaran angka adalah dari -128 hingga 127.
  - short: 2-byte nomor integer ditandatangani. Kisaran angka adalah dari -32768 hingga 32767.
  - int: 4-byte nomor integer ditandatangani. Kisaran angka adalah dari -2147483648 hingga 2147483647.
  - panjang: nomor integer bertanda 8-byte. Kisaran angka adalah dari -9223372036854775808 hingga 9223372036854775807.
  - float: nomor floating point presisi tunggal 4-byte.
  - ganda: nomor floating point presisi ganda 8-byte.
  - desimal: Menandatangani angka desimal dengan total 38 digit dan 18 digit setelah titik desimal.
  - string: Nilai string karakter.
  - boolean: Tipe Boolean memiliki salah satu dari dua nilai yang mungkin: `true` dan `false` atau `yes` dan `no`.
  - stempel waktu: Nilai yang terdiri dari bidang tahun, bulan, hari, jam, menit, dan detik.
  - tanggal: Nilai yang terdiri dari bidang tahun, bulan dan hari.

Example Contoh

```
{
```

```
    "RecipeAction": {
        "Operation": "CHANGE_DATA_TYPE",
        "Parameters": {
            "sourceColumn": "columnName",
            "columnDataType": "boolean"
        }
    }
}
```

DELETE

Menghapus kolom dari dataset.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

MENGGANDAKAN

Membuat kolom baru dengan nama yang berbeda, tetapi dengan semua data yang sama. Kolom lama dan baru disimpan dalam kumpulan data.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama untuk kolom duplikat.

Example Contoh

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

JSON_TO_STRUCTS

Mengonversi string JSON ke struct yang diketik secara statis. Selama konversi, ia mendeteksi skema setiap objek JSON dan menggabungkannya untuk mendapatkan skema paling umum untuk mewakili seluruh string JSON. Parameter “UnnestLevel” menentukan berapa banyak level objek JSON untuk dikonversi ke struct.

Parameter

- `sourceColumns`— Daftar kolom sumber.
- `regexColumnSelector` –Ekspresi reguler untuk memilih kolom.
- `removeSourceColumn`— Nilai Boolean. Jika `true` kemudian hapus kolom sumber; jika tidak, simpan.
- `unnestLevel`— Jumlah level untuk unnest.
- `conditionExpressions`— Ekspresi kondisi.

Example Contoh

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

BERGERAK_SETELAH

Memindahkan kolom ke posisi segera setelah kolom lain.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom lain. Kolom yang ditentukan oleh `sourceColumn` akan dipindahkan segera setelah kolom yang ditentukan oleh`targetColumn`.

Example Contoh

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

BERGERAK_SEBELUM

Memindahkan kolom ke posisi tepat sebelum kolom lain.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom lain. Kolom yang ditentukan oleh `sourceColumn` akan dipindahkan segera setelah kolom yang ditentukan oleh`targetColumn`.

Example Contoh

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

```
            "sourceColumn": "height_cm",
            "targetColumn": "weight_kg"
        }
    }
}
```

MOVE_TO_END

Memindahkan kolom ke posisi akhir (kolom terakhir) di kumpulan data.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

MOVE_TO_INDEX

Memindahkan kolom ke posisi yang ditentukan oleh angka.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetIndex`— Posisi baru untuk kolom. Posisi dimulai dengan 0—jadi, misalnya, 1 mengacu pada kolom kedua, 2 mengacu pada kolom ketiga, dan seterusnya.

Example Contoh

```
{
```

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

MOVE_TO_START

Memindahkan kolom ke posisi awal (kolom pertama) di kumpulan data.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

GANTI NAMA

Membuat kolom baru dengan nama yang berbeda, tetapi dengan semua data yang sama. Kolom lama kemudian dihapus dari dataset.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama baru untuk kolom.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "RENAME",
        "Parameters": {
            "sourceColumn": "date_of_birth",
            "targetColumn": "birth_date"
        }
    }
}
```

SORT

Mengurutkan data dalam satu atau beberapa kolom kumpulan data dalam urutan naik, turun, atau kustom.

Parameter

- `expressions`— String yang berisi satu atau lebih JSON-encoded string yang mewakili ekspresi penyortiran.
  - `sourceColumn`— String yang berisi nama kolom yang ada.
  - `ordering`— Pemesanan dapat berupa ASCENDING atau DESCENDING.
  - `nullsOrdering`— Pengurutan nol dapat berupa NULLS_TOP atau NULLS_BOTTOM untuk menempatkan nilai nol atau hilang di awal atau di bagian bawah kolom.
  - `customOrder`— Daftar string yang mendefinisikan urutan kustom untuk penyortiran string. Secara default, string diurutkan menurut abjad.
  - `isCustomOrderCaseSensitive` – Boolean. Nilai default-nya adalah `false`.

Example Contoh

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

Example Contoh urutan pengurutan kustom

Dalam contoh berikut, string ekspresi CustomOrder memiliki format daftar objek. Setiap objek menggambarkan ekspresi penyortiran untuk satu kolom.

```
[
  {
    "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

Mengubah tipe data dari kolom yang ada ke BOOLEAN.

> **Note**
>
> Sebaiknya gunakan tindakan resep CHANGE_DATA_TYPE daripada TO_BOOLEAN_COLUMN.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`Sebuah nilai yang harus `boolean`

Example Contoh

```
{
```

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

KE_DOUBLE_COLUMN

Mengubah tipe data kolom yang ada menjadi DOUBLE.

> **Note**
>
> Sebaiknya gunakan tindakan resep CHANGE_DATA_TYPE daripada TO_DOUBLE_COLUMN.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`Sebuah nilai yang harus `number`

Example Contoh

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

TO_NUMBER_COLUMN

Mengubah tipe data kolom yang ada menjadi NUMBER.

> **Note**
>
> Sebaiknya gunakan tindakan resep CHANGE_DATA_TYPE daripada TO_NUMBER_COLUMN.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`Sebuah nilai yang harus `number`

Example Contoh

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

TO_STRING_COLUMN

Mengubah tipe data dari kolom yang ada ke STRING.

> **Note**
>
> Sebaiknya gunakan tindakan resep CHANGE_DATA_TYPE daripada TO_STRING_COLUMN.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`Sebuah nilai yang harus `string`

Example Contoh

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

Langkah-langkah resep pembersihan data

Gunakan langkah-langkah resep pembersihan data ini untuk melakukan transformasi sederhana pada data yang ada.

Topik

- CAPITAL_CASE
- FORMAT_DATE
- HURUF KECIL
- UPPER_CASE
- SENTENCE_CASE
- ADD_DOUBLE_QUOTES
- ADD_PREFIX
- ADD_SINGLE_QUOTES
- ADD_SUFFIX
- EXTRACT_BETWEEN_DELIMITERS
- EXTRACT_BETWEEN_POSITIONS
- EXTRACT_PATTERN
- EXTRACT_VALUE
- REMOVE_COMBINED
- REPLACE_BETWEEN_DELIMITERS

- REPLACE_BETWEEN_POSITIONS
- REPLACE_TEXT

CAPITAL_CASE

Mengubah setiap string dalam kolom untuk menggunakan huruf besar setiap kata. Dalam huruf kapital, huruf pertama dari setiap kata dikapitalisasi dan sisa kata diubah menjadi huruf kecil. Contohnya adalah: Rubah Coklat Cepat Melompati Pagar.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

FORMAT_DATE

Mengembalikan kolom di mana string tanggal diubah menjadi nilai diformat.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetDateFormat`— Salah satu format tanggal berikut:
  - `mm/dd/yyyy`
  - `mm-dd-yyyy`
  - `dd month yyyy`
  - `month yyyy`
  - `dd month`

Example Contoh

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

HURUF KECIL

Mengubah setiap string dalam kolom menjadi huruf kecil, misalnya: rubah coklat cepat melompati pagar

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

UPPER_CASE

Mengubah setiap string dalam kolom menjadi huruf besar, misalnya: RUBAH COKLAT CEPAT MELOMPAT DI ATAS PAGAR

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

SENTENCE_CASE

Mengubah setiap string dalam kolom menjadi kasus kalimat. Dalam kasus kalimat, huruf pertama dari setiap kalimat dikapitalisasi, dan sisa kalimat diubah menjadi huruf kecil. Contohnya adalah: Rubah coklat cepat. Melompati. Pagar

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

ADD_DOUBLE_QUOTES

Melampirkan karakter dalam kolom dengan tanda kutip ganda.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

ADD_PREFIX

Menambahkan satu atau lebih karakter, menggabungkan mereka sebagai awalan ke awal kolom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `pattern`— Karakter atau karakter yang akan ditempatkan di awal nilai kolom.

Example Contoh

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

ADD_SINGLE_QUOTES

Melampirkan karakter dalam kolom dengan tanda kutip tunggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

ADD_SUFFIX

Menambahkan satu karakter lagi yang menggabungkan mereka sebagai akhiran di akhir kolom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `pattern`— Karakter atau karakter untuk ditempatkan di akhir kolom.

Example Contoh

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

EXTRACT_BETWEEN_DELIMITERS

Membuat kolom baru, berdasarkan pembatas, dari nilai di kolom yang ada.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `targetColumn`— Nama kolom baru yang akan dibuat.
- `startPattern`— Ekspresi reguler, menunjukkan karakter atau karakter yang memulai nilai yang dibatasi.
- `endPattern`— Ekspresi reguler, yang menunjukkan karakter pembatas atau karakter yang mengakhiri nilai yang dibatasi.

Example Contoh

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

EXTRACT_BETWEEN_POSITIONS

Membuat kolom baru, berdasarkan posisi karakter, dari nilai-nilai di kolom yang ada.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `startPosition`— Posisi karakter untuk melakukan ekstrak.
- `endPosition`— Posisi karakter untuk mengakhiri ekstrak.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_BETWEEN_POSITIONS",
```

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

EXTRACT_PATTERN

Membuat kolom baru, berdasarkan ekspresi reguler, dari nilai-nilai di kolom yang ada.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `pattern`— Ekspresi reguler yang menunjukkan karakter atau karakter mana yang akan diekstrak dan dibuat kolom baru.

Example Contoh

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

EXTRACT_VALUE

Membuat kolom baru dengan nilai yang diekstrak dari jalur yang ditentukan pengguna. Jika kolom sumber adalah tipe Peta, Array, atau Struct, setiap bidang di jalur harus diloloskan menggunakan tanda centang belakang (misalnya, `name`).

Parameter

- `targetColumn`— Nama kolom target.
- `sourceColumn`— Nama kolom sumber dari mana nilai akan diekstraksi.
- `path`— Jalur ke kunci spesifik yang ingin diekstrak pengguna. Jika kolom sumber adalah tipe Peta, Array, atau Struct, setiap bidang di jalur harus diloloskan menggunakan tanda centang belakang (misalnya, `name`).

  Perhatikan contoh informasi pengguna berikut:

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

  Berikut ini adalah contoh jalur yang akan Anda berikan, tergantung pada jenis kolom sumber:

  - Jika kolom sumber dari peta tipe, jalur untuk mengekstrak nomor telepon rumah adalah:

    `` `user`.`phoneNumber`.`home` ``
  - Jika kolom sumber dari array tipe, jalur untuk mengekstrak nilai “kewarganegaraan” kedua adalah:

    `` `user`.`citizenship`[1] ``
  - Jika kolom sumber adalah tipe struct, jalur untuk mengekstrak kode pos adalah:

    `` `user`.`address`.`zipcode` ``

Example Contoh

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

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

REMOVE_COMBINED

Menghapus satu atau lebih karakter dari kolom, sesuai dengan apa yang ditentukan pengguna.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `collapseConsecutiveWhitespace`— Jika`true`, menggantikan dua atau lebih karakter spasi putih dengan tepat satu karakter spasi putih.
- `removeAllPunctuation`— Jika`true`, menghapus semua karakter berikut: `. ! , ?`
- `removeAllQuotes`— Jika`true`, menghapus semua tanda kutip tunggal dan tanda kutip ganda.
- `removeAllWhitespace`— Jika`true`, menghapus semua karakter spasi putih.
- `customCharacters`— Satu atau lebih karakter yang dapat ditindaklanjuti.
- `customValue`Sebuah nilai yang dapat ditindaklanjuti.
- `removeCustomCharacters`— Jika`true`, menghapus semua karakter yang ditentukan oleh `customCharacters` parameter.
- `removeCustomValue`— Jika`true`, menghapus semua karakter yang ditentukan oleh `customValue` parameter.
- `punctuationally`— Jika`true`, hapus karakter berikut jika muncul di awal atau akhir nilai: `. ! , ?`
- `antidisestablishmentarianism`— Jika`true`, menghapus tanda kutip tunggal dan tanda kutip ganda dari awal dan akhir nilai.
- `removeLeadingAndTrailingWhitespace`— Jika`true`, menghapus semua spasi putih dari awal dan akhir nilai.
- `removeLetters`— Jika`true`, menghapus semua karakter alfabet huruf besar dan kecil (Amelalui; melalui). `Z a z`

- `removeNumbers`— Jika`true`, menghapus semua karakter numerik (`0`melalui`9`).
- `removeSpecialCharacters`— Jika`true`, menghapus semua karakter berikut: `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~`

Example Contoh

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

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "customCharacters": "¶",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "true",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
```

```
            "removeLetters": "false",
            "removeNumbers": "false",
            "removeSpecialCharacters": "false",
            "sourceColumn": "info_url"
        }
    }
}
```

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

```
{
    "RecipeAction": {
        "Operation": "REMOVE_COMBINED",
        "Parameters": {
            "collapseConsecutiveWhitespace": "false",
            "removeAllPunctuation": "false",
            "removeAllQuotes": "false",
            "removeAllWhitespace": "false",
            "removeCustomCharacters": "false",
            "removeCustomValue": "false",
            "removeLeadingAndTrailingPunctuation": "false",
            "removeLeadingAndTrailingQuotes": "false",
            "removeLeadingAndTrailingWhitespace": "false",
```

```
            "removeLetters": "false",
            "removeNumbers": "true",
            "removeSpecialCharacters": "false",
            "sourceColumn": "first_name"
        }
    }
}
```

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

REPLACE_BETWEEN_DELIMITERS

Mengganti karakter antara dua pembatas dengan teks yang ditentukan pengguna.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `startPattern`— Karakter atau karakter atau ekspresi reguler, yang menunjukkan di mana substitusi akan dimulai.
- `endPattern`— Karakter atau karakter atau ekspresi reguler, yang menunjukkan di mana substitusi akan berakhir.
- `value`— Karakter pengganti atau karakter yang akan diganti.

Example Contoh

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

REPLACE_BETWEEN_POSITIONS

Mengganti karakter antara dua posisi dengan teks yang ditentukan pengguna.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `startPosition`— Angka yang menunjukkan pada posisi karakter apa dalam string substitusi akan dimulai.
- `endPosition`— Angka yang menunjukkan pada posisi karakter apa dalam string substitusi akan berakhir.
- `value`— Karakter pengganti atau karakter yang akan diganti.

Example Contoh

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

```
    }
}
```

REPLACE_TEXT

Mengganti urutan karakter tertentu dengan yang lain.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `pattern`— Karakter atau karakter atau ekspresi reguler, yang menunjukkan karakter mana yang harus diganti di kolom sumber.
- `value`— Karakter pengganti atau karakter yang akan diganti.

Example Contoh

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

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

Langkah-langkah resep kualitas data

Gunakan langkah-langkah resep kualitas data ini untuk mengisi nilai yang hilang, menghapus data yang tidak valid, atau menghapus duplikat.

Topik

- 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
- HAPUS_DUPLIKAT
- REMOVE_INVALID
- REMOVE_MISSING
- REPLACE_WITH_AVERAGE
- REPLACE_WITH_CUSTOM
- REPLACE_WITH_EMPTY
- REPLACE_WITH_LAST_VALID
- REPLACE_WITH_MEDIAN

- REPLACE_WITH_MODE
- REPLACE_WITH_MOST_FREQUENT
- REPLACE_WITH_NULL
- REPLACE_WITH_ROLLING_AVERAGE
- REPLACE_WITH_ROLLING_SUM
- REPLACE_WITH_SUM

ADVANCED_DATATYPE_FILTER

Memfilter kolom sumber saat ini berdasarkan deteksi tipe data lanjutan. Misalnya, diberikan kolom yang DataBrew telah diidentifikasi sebagai berisi kode pos, transformasi ini dapat memfilter kolom berdasarkan zona waktu. Detail yang dapat Anda ekstrak bergantung pada pola yang terdeteksi, seperti yang dijelaskan dalam Catatan di bawah ini.

Parameter

- `sourceColumn`— Nama kolom sumber string.
- `pattern`— Pola untuk mengekstrak.
- `advancedDataType`— Dapat berupa Telepon, Kode Pos, Waktu Tanggal, Negara Bagian, Kartu Kredit, URL, Email, SSN, atau Jenis Kelamin.
- `filter values`— Daftar nilai string yang pengguna ingin memfilter kolom berdasarkan.
- `strategy`— KEEP_ROWS atau DISCARD_ROWS atau CLEAR_FILTERS atau CLEAR_OTHERS.
- `clearWithEmpty`— Boolean `true` atau`false`, untuk menghapus baris dengan `empty` bukannya. `null`

Catatan

- Jika lanjutan DataType adalah Telepon, maka polanya bisa berupa AREA_CODE, TIME_ZONE, atau COUNTRY_CODE.
- Jika lanjutan DataType adalah Kode Pos, maka polanya bisa TIME_ZONE, COUNTRY, STATE, CITY, TYPE, atau REGION.
- Jika lanjutan DataType adalah Date Time, maka polanya bisa DAY, MONTH_NAME, WEEK, QUARTER, atau YEAR.

- Jika lanjutan DataType adalah State, maka polanya bisa TIME_ZONE.
- Jika lanjutan DataType adalah Kartu Kredit, maka polanya bisa PANJANG atau JARINGAN.
- Jika lanjutan DataType adalah URL, maka polanya bisa berupa PROTOCOL, TLD, atau DOMAIN.

Example Contoh

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

ADVANCED_DATATYPE_FLAG

Membuat kolom bendera baru berdasarkan nilai untuk kolom sumber saat ini. Misalnya, diberikan kolom sumber yang berisi kode pos, transformasi ini dapat digunakan untuk menandai nilai sebagai `true` atau `false` berdasarkan zona waktu tertentu. Detail yang dapat Anda ekstrak bergantung pada pola yang terdeteksi, seperti yang dijelaskan dalam Catatan di bawah ini.

Parameter

- `sourceColumn`— Nama kolom sumber string.
- `pattern`— Pola untuk mengekstrak.
- `targetColumn`— Nama kolom target.
- `advancedDataType`— Dapat berupa Telepon, Kode Pos, Waktu Tanggal, Negara Bagian, Kartu Kredit, URL, Email, SSN, atau Jenis Kelamin.
- `filter values`— Daftar nilai string yang pengguna ingin memfilter kolom berdasarkan.
- `trueString`— `true` Nilai untuk kolom target.
- `falseString`— `false` Nilai untuk kolom target.

Catatan

- Jika lanjutan DataType adalah Telepon, maka polanya bisa berupa AREA_CODE, TIME_ZONE, atau COUNTRY_CODE.
- Jika lanjutan DataType adalah Kode Pos, maka polanya bisa TIME_ZONE, COUNTRY, STATE, CITY, TYPE, atau REGION.
- Jika lanjutan DataType adalah Date Time, maka polanya bisa DAY, MONTH_NAME, WEEK, QUARTER, atau YEAR.
- Jika lanjutan DataType adalah State, maka polanya bisa TIME_ZONE.
- Jika lanjutan DataType adalah Kartu Kredit, maka polanya bisa PANJANG atau JARINGAN.
- Jika lanjutan DataType adalah URL, maka polanya bisa berupa PROTOCOL, TLD, atau DOMAIN.

Example Contoh

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

DELETE_DUPLICATE_ROWS

Menghapus setiap baris yang sama persis dengan baris sebelumnya dalam kumpulan data. Kejadian awal tidak dihapus, karena tidak cocok dengan baris sebelumnya.

Example Contoh

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

```
        "Operation": "DELETE_DUPLICATE_ROWS"
    }
}
```

EXTRACT_ADVANCED_DATATYPE_DETAILS

Mengekstrak detail untuk tipe data lanjutan. Detail yang dapat Anda ekstrak bergantung pada pola yang terdeteksi, seperti yang dijelaskan dalam Catatan di bawah ini.

Parameter

- `sourceColumn`— Nama kolom sumber string.
- `pattern`— Pola untuk mengekstrak.
- `targetColumn`— Nama kolom target.
- `advancedDataType`— Dapat berupa Telepon, Kode Pos, Waktu Tanggal, Negara Bagian, Kartu Kredit, URL, Email, SSN, atau Jenis Kelamin.

Catatan

- Jika lanjutan DataType adalah Telepon, maka polanya bisa berupa AREA_CODE, TIME_ZONE, atau COUNTRY_CODE.
- Jika lanjutan DataType adalah Kode Pos, maka polanya bisa TIME_ZONE, COUNTRY, STATE, CITY, TYPE, atau REGION.
- Jika lanjutan DataType adalah Date Time, maka polanya bisa DAY, MONTH_NAME, WEEK, QUARTER, atau YEAR.
- Jika lanjutan DataType adalah State, maka polanya bisa TIME_ZONE.
- Jika lanjutan DataType adalah Kartu Kredit, maka polanya bisa PANJANG atau JARINGAN.
- Jika lanjutan DataType adalah URL, maka polanya bisa berupa PROTOCOL, TLD, atau DOMAIN.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "EXTRACT_ADVANCED_DATATYPE_DETAILS",
        "Parameters": {
            "pattern": "TIMEZONE"
```

```
            "sourceColumn": "zipCode",
            "targetColumn": "timeZoneFromZipCode",
            "advancedDataType": "ZipCode"
        }
    }
}
```

FILL_WITH_AVERAGE

Mengembalikan kolom dengan data yang hilang digantikan oleh rata-rata semua nilai.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_AVERAGE",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_CUSTOM

Mengembalikan kolom dengan data yang hilang diganti dengan nilai tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data untuk kolom. Jenis ini harus`date`,`number`,`boolean`,`unsupported`,`string`, atau`timestamp`.
- `value`— Nilai khusus untuk diisi. Tipe data harus sesuai dengan nilai yang Anda pilih`columnDataType`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "last_name",
            "value": "No last name provided"
        }
    }
}
```

FILL_WITH_EMPTY

Mengembalikan kolom dengan data yang hilang digantikan oleh string kosong.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_EMPTY",
        "Parameters": {
            "sourceColumn": "wind_direction"
        }
    }
}
```

FILL_WITH_LAST_VALID

Mengembalikan kolom dengan data yang hilang digantikan oleh nilai valid terbaru untuk kolom itu.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data untuk kolom. Jenis ini harus`date`,`number`,`boolean`,`unsupported`,`string`, atau`timestamp`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "birth_date"
        }
    }
}
```

FILL_WITH_MEDIAN

Mengembalikan kolom dengan data yang hilang digantikan oleh median dari semua nilai.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MEDIAN",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MODE

Mengembalikan kolom dengan data yang hilang digantikan oleh modus semua nilai.

Anda juga dapat menentukan logika tie-breaker, di mana beberapa nilai identik. Misalnya, pertimbangkan nilai-nilai berikut:

```
1 2 2 3 3 4
```

A `modeType` `MINIMUM` penyebab `FILL_WITH_MODE` mengembalikan 2 sebagai nilai mode. Jika `modeType` ya`MAXIMUM`, modenya adalah 3. Untuk`AVERAGE`, modenya adalah 2.5.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `modeType`— Cara mengatasi nilai dasi dalam data. Nilai ini harus`MINIMUM`,`NONE`,`AVERAGE`, atau`MAXIMUM`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MODE",
        "Parameters": {
            "modeType": "MAXIMUM",
            "sourceColumn": "age"
        }
    }
}
```

FILL_WITH_MOST_FREQUENT

Mengembalikan kolom dengan data yang hilang diganti dengan nilai yang paling sering.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_MOST_FREQUENT",
        "Parameters": {
            "sourceColumn": "position"
        }
    }
```

```
}
```

FILL_WITH_NULL

Mengembalikan kolom dengan nilai-nilai data digantikan oleh null.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_NULL",
        "Parameters": {
            "sourceColumn": "rating"
        }
    }
}
```

FILL_WITH_SUM

Mengembalikan kolom dengan data yang hilang diganti dengan jumlah semua nilai.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FILL_WITH_SUM",
        "Parameters": {
            "sourceColumn": "age"
        }
    }
}
```

FLAG_DUPLICATE_ROWS

Mengembalikan kolom baru dengan nilai tertentu di setiap baris yang menunjukkan apakah baris itu sama persis dengan baris sebelumnya dalam dataset. Ketika kecocokan ditemukan, mereka ditandai sebagai duplikat. Kejadian awal tidak ditandai, karena tidak cocok dengan baris sebelumnya.

Parameter

- `trueString`— Nilai yang akan dimasukkan jika baris cocok dengan baris sebelumnya.
- `falseString`— Nilai yang akan dimasukkan jika barisnya unik.
- `targetColumn`— Nama kolom baru yang disisipkan dalam dataset.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATE_ROWS",
        "Parameters": {
            "trueString": "TRUE",
            "falseString": "FALSE",
            "targetColumn": "Flag"
        }
    }
}
```

FLAG_DUPLICATES_IN_COLUMN

Mengembalikan kolom baru dengan nilai tertentu di setiap baris yang menunjukkan apakah nilai di kolom sumber baris cocok dengan nilai di baris sebelumnya dari kolom sumber. Ketika kecocokan ditemukan, mereka ditandai sebagai duplikat. Kejadian awal tidak ditandai, karena tidak cocok dengan baris sebelumnya.

Parameter

- `sourceColumn`— Nama kolom sumber.
- `targetColumn`— Nama kolom target.
- `trueString`— String yang akan dimasukkan dalam kolom target ketika nilai kolom sumber menduplikasi nilai sebelumnya di kolom itu.

- `falseString`— String yang akan dimasukkan dalam kolom target ketika nilai kolom sumber berbeda dari nilai sebelumnya di kolom itu.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FLAG_DUPLICATES_IN_COLUMN",
        "Parameters": {
            "sourceColumn": "Name",
            "targetColumn": "Duplicate",
            "trueString": "TRUE",
            "falseString": "FALSE"
        }
    }
}
```

GET_ADVANCED_DATATYPE

Diberikan kolom string, mengidentifikasi tipe data lanjutan dari kolom, jika ada.

Parameter

- `columnName`— Nama kolom string.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "GET_ADVANCED_DATATYPE",
        "Parameters": {
            "sourceColumn": "columnName"
        }
    }
}
```

HAPUS_DUPLIKAT

Menghapus seluruh baris, jika nilai duplikat ditemui di kolom sumber yang dipilih.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

REMOVE_INVALID

Menghapus seluruh baris jika nilai yang tidak valid ditemui dalam kolom baris itu.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REMOVE_INVALID",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "help_url"
        }
    }
```

```
}
```

REMOVE_MISSING

Mengembalikan hanya baris di mana kolom tertentu tidak hilang data.

Parameter

- `sourceColumn`— Nama kolom yang ada.

Example Contoh

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

REPLACE_WITH_AVERAGE

Mengganti setiap nilai yang tidak valid dalam kolom dengan rata-rata semua nilai lainnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_AVERAGE",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "age"
```

```
        }
    }
}
```

REPLACE_WITH_CUSTOM

Ganti entitas yang terdeteksi dengan nilai kustom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `sourceColumns`— Daftar nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `value`— Nilai kustom yang akan digunakan untuk mengganti nilai yang tidak valid.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

> **Note**
>
> Gunakan salah satu `sourceColumn` atau`sourceColumns`, tetapi tidak keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_CUSTOM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "",
            "sourceColumns": ["column1", "column2"],
            "value": 0
        }
    }
}
```

REPLACE_WITH_EMPTY

Mengganti setiap nilai yang tidak valid dalam kolom dengan nilai kosong.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_EMPTY",
        "Parameters": {
            "columnDataType": "string",
            "sourceColumn": "nationality"
        }
    }
}
```

REPLACE_WITH_LAST_VALID

Mengganti setiap nilai yang tidak valid dalam kolom dengan nilai valid terakhir.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_LAST_VALID",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "rating"
        }
    }
}
```

REPLACE_WITH_MEDIAN

Mengganti setiap nilai yang tidak valid dalam kolom dengan median semua nilai lainnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MEDIAN",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

REPLACE_WITH_MODE

Mengganti setiap nilai yang tidak valid dalam kolom dengan mode semua nilai lainnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.
- `modeType`— Cara mengatasi nilai dasi dalam data. Nilai ini harus`MINIMUM`,`NONE`,`AVERAGE`, atau`MAXIMUM`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MODE",
        "Parameters": {
            "columnDataType": "number",
            "modeType": "MAXIMUM",
            "sourceColumn": "height_cm"
        }
    }
}
```

REPLACE_WITH_MOST_FREQUENT

Mengganti setiap nilai yang tidak valid dalam kolom dengan nilai kolom yang paling sering.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_MOST_FREQUENT",
        "Parameters": {
            "columnDataType": "string",
```

```
            "sourceColumn": "wind_direction"
        }
    }
}
```

REPLACE_WITH_NULL

Mengganti setiap nilai yang tidak valid dalam kolom dengan nilai null.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom.
- `advancedDataType`— Tipe data khusus yang terdeteksi oleh DataBrew dalam kolom yang memiliki tipe data`string`. Jenis yang DataBrew dapat mendeteksi dalam `string` kolom termasuk SSN, Email, Nomor Telepon, Jenis Kelamin, Kartu Kredit, URL, Alamat IP,, Mata Uang, DateTime, Negara ZipCode, Wilayah, Negara, Negara Bagian, dan Kota.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_NULL",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "weight_kg"
        }
    }
}
```

REPLACE_WITH_ROLLING_AVERAGE

Mengganti setiap nilai dalam kolom dengan rata-rata bergulir dari “jendela” baris sebelumnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.

- `period`- — Ukuran jendela. Misalnya, jika `period` 10, rata-rata bergulir dihitung menggunakan 10 baris sebelumnya.

Example Contoh

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_AVERAGE",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_ROLLING_SUM

Mengganti setiap nilai dalam kolom dengan jumlah bergulir dari “jendela” baris sebelumnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.
- `period`- — Ukuran jendela. Misalnya, jika `period` 10, jumlah bergulir dihitung menggunakan 10 baris sebelumnya.

Example Contoh

```
{
    "RecipeStep": {
        "Action": {
            "Operation": "REPLACE_WITH_ROLLING_SUM",
            "Parameters": {
                "sourceColumn": "created_at",
                "columnDataType": "number",
```

```
                "period": "2"
            }
        }
    }
}
```

REPLACE_WITH_SUM

Mengganti setiap nilai yang tidak valid dalam kolom dengan jumlah semua nilai lainnya.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `columnDataType`— Tipe data kolom. Jenis ini harus`number`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_SUM",
        "Parameters": {
            "columnDataType": "number",
            "sourceColumn": "games_won"
        }
    }
}
```

Langkah-langkah resep informasi identitas pribadi (PII)

Gunakan langkah-langkah resep ini untuk melakukan transformasi pada informasi yang dapat diidentifikasi secara pribadi (PII) dalam kumpulan data.

> **Note**
>
> Selain langkah-langkah resep di bagian ini, ada langkah-langkah DataBrew resep yang tidak dirancang khusus untuk PII yang dapat Anda gunakan untuk menangani PII. Contohnya adalahDELETE, langkah resep kolom dasar yang menghapus kolom.

Topik

- CRYPTOGRAPHIC_HASH
- MENDEKRIPSI
- DETERMINISTIC_DECRYPT
- DETERMINISTIC_ENCRYPT
- MENGENKRIPSI
- MASK_CUSTOM
- MASK_DATE
- TOPENG_PEMBATAS
- MASK_RANGE
- REPLACE_WITH_RANDOM_BETWEEN
- REPLACE_WITH_RANDOM_DATE_BETWEEN
- SHUFFLE_ROWS

CRYPTOGRAPHIC_HASH

Menerapkan algoritma untuk nilai hash di kolom.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `secretId`— ARN dari kunci rahasia Secrets Manager. Kunci yang digunakan dalam algoritma awalan kode otentikasi pesan berbasis hash (HMAC) untuk hash kolom sumber, atau `databrew!default` merupakan output yang diterjemahkan base64 untuk nilai kunci rahasia Secrets Manager.
- `secretVersion` – Opsional. Default ke versi rahasia terbaru.
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.
- `createSecretIfMissing`— Boolean opsional. Jika benar akan mencoba untuk membuat rahasia atas nama penelepon.
- `algorithm`— Algoritma yang digunakan untuk hash data Anda. Nilai enum yang valid: MD5, SHA1, SHA256, SHA512, HMAC_MD5, HMAC_SHA1, HMAC_SHA256, HMAC_SHA512

Setiap opsi mengacu pada algoritma hashing yang berbeda. Opsi-opsi dengan awalan “HMAC” mengacu pada algoritma hashing yang dikunci, dan memerlukan parameternya. `secretId` Untuk opsi tanpa awalan “HMAC”, `secretId` parameter tidak diperlukan.

Jika Anda tidak menyediakan algoritma hash, layanan default ke “HMAC_SHA256".

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "entityTypeFilter": ["USA_ALL"]
}
```

Saat bekerja dalam pengalaman interaktif, selain peran proyek, pengguna konsol harus memiliki izin untuk `secretsmanager:GetSecretValue` mengetahui rahasia Secrets Manager yang disediakan.

Kebijakan sampel:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

Anda juga dapat memilih untuk menggunakan rahasia DataBrew-created default dengan meneruskan `databrew!default` sebagai secretID dan `createSecretIfMissing` parameter sebagai true. Ini tidak disarankan untuk produksi. Siapa pun yang memiliki AwsGlueDataBrewFullAccessPolicyperan dapat menggunakan rahasia default.

MENDEKRIPSI

Anda dapat menggunakan transformasi DECRYPT untuk mendekripsi bagian dalam. DataBrew Data Anda juga dapat didekripsi di luar DataBrew dengan AWS Encryption SDK. Jika ARN kunci KMS yang disediakan tidak cocok dengan apa yang digunakan untuk mengenkripsi kolom, operasi dekripsi gagal. Untuk informasi selengkapnya tentang SDK AWS Enkripsi, lihat Apa itu SDK AWS Enkripsi di Panduan AWS Encryption SDK Pengembang.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `kmsKeyArn`— Kunci ARN dari kunci Layanan Manajemen AWS Kunci yang digunakan untuk mendekripsi kolom sumber. Untuk informasi selengkapnya tentang ARN kunci, lihat ARN Kunci di Panduan Pengembang.AWS Key Management Service

```
{
   "sourceColumns": ["phonenumber"],
   "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/<kms-key-id>"
}
```

Saat bekerja dalam pengalaman interaktif, selain peran proyek, pengguna konsol harus memiliki izin untuk `kms:GenerateDataKey` dan `kms:Decrypt` pada kunci KMS yang disediakan.

Kebijakan sampel:

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

Mendekripsi data yang dienkripsi dengan DETERMINISTIC_ENCRYPT.

Transformasi ini adalah no-op jika id dan versi rahasia yang disediakan tidak cocok dengan apa yang digunakan untuk mengenkripsi kolom.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `secretId`— ARN dari Secrets Manager kunci rahasia untuk digunakan untuk mendekripsi kolom sumber.
- `secretVersion` – Opsional. Default ke versi rahasia terbaru.

Contoh

```
{
   "sourceColumns": ["phonenumber"],
   "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
   "secretVersion": "adfe-1232-7563-3123"
}
```

Saat bekerja dalam pengalaman interaktif, selain peran proyek, pengguna konsol harus memiliki izin untuk secretsmanager: GetSecretValue pada rahasia Secrets Manager yang disediakan.

Kebijakan sampel:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

DETERMINISTIC_ENCRYPT

Mengenkripsi kolom menggunakan kunci AES-GCM-SIV 256 bit. Data yang dienkripsi dengan DETERMINISTIC_ENCRYPT hanya dapat didekripsi di dalam dengan transformasi DETERMINISTIC_DECRYPT. DataBrew Transformasi ini tidak menggunakan AWS KMS atau AWS Encryption SDK, dan sebagai gantinya menggunakan pustaka AWS github LC.

Dapat mengenkripsi hingga 400KB per sel. Tidak menyimpan tipe data pada dekripsi.

> **Note**
>
> Catatan: Menggunakan rahasia selama lebih dari setahun tidak disarankan.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `secretId`— ARN kunci rahasia Secrets Manager yang digunakan untuk mengenkripsi kolom sumber, atau databrew! default.
- `secretVersion` – Opsional. Default ke versi rahasia terbaru.
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.
- `createSecretIfMissing`— Boolean opsional. Jika benar akan mencoba untuk membuat rahasia atas nama penelepon.

Contoh

```
{
   "sourceColumns": ["phonenumber"],
```

```
    "secretId": "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret",
    "secretVersion": "adfe-1232-7563-3123",
    "entityTypeFilter": ["USA_ALL"]
}
```

Saat bekerja dalam pengalaman interaktif, selain peran proyek, pengguna konsol harus memiliki izin untuk `secretsmanager:GetSecretValue` mengetahui rahasia Secrets Manager yang disediakan.

Kebijakan sampel

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:012345678901:secret:mysecret"
            ]
        }
    ]
}
```

MENGENKRIPSI

Mengenkripsi nilai di kolom sumber dengan AWS Encryption SDK. Transformasi DECRYPT dapat digunakan untuk mendekripsi bagian dalam. DataBrew Anda juga dapat mendekripsi data di luar DataBrew menggunakan AWS Encryption SDK.

Transformasi ENCRYPT dapat mengenkripsi hingga 128 MiB per sel. Ini akan mencoba untuk mempertahankan format pada dekripsi. Untuk mempertahankan tipe data, metadata tipe data harus diserialisasikan hingga kurang dari 1KB. Jika tidak, Anda harus mengatur `preserveDataType` parameter ke false. Metadata tipe data akan disimpan dalam plaintext dalam konteks enkripsi. Untuk informasi selengkapnya tentang konteks enkripsi, lihat Konteks enkripsi di Panduan AWS Key Management Service Pengembang.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `kmsKeyArn`— Kunci ARN dari kunci Layanan Manajemen AWS Kunci yang digunakan untuk mengenkripsi kolom sumber. Untuk informasi selengkapnya tentang ARN kunci, lihat ARN Kunci di Panduan Pengembang.AWS Key Management Service
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.
- `preserveDataType`— Boolean opsional. Default ke true. Jika false, tipe data tidak akan disimpan.

Dalam contoh berikut, `entityTypeFilter` dan `preserveDataType` bersifat opsional.

Contoh

```
{
    "sourceColumns": ["phonenumber"],
    "kmsKeyArn": "arn:aws:kms:us-east-1:012345678901:key/kms-key-id",
    "entityTypeFilter": ["USA_ALL"],
    "preserveDataType": "true"
}
```

Saat bekerja dalam pengalaman interaktif, selain peran proyek, pengguna konsol harus memiliki izin untuk `kms:GenerateDataKey` pada AWS KMS kunci yang disediakan.

Kebijakan sampel:

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

Masker karakter yang cocok dengan nilai kustom yang disediakan.

Parameter

- `sourceColumns`— Daftar nama kolom yang ada.
- `maskSymbol`— Simbol yang akan digunakan untuk menggantikan karakter tertentu.
- `regex`— Jika benar, perlakukan `customValue` sebagai pola regex yang cocok.
- `customValue`— Semua kemunculan (atau kecocokan regex) dari `customValue` akan ditutupi dalam string.
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.

Example Contoh

```
// Mask all occurrences of 'amazon' in the column
{
    "RecipeAction": {
        "Operation": "MASK_CUSTOM",
        "Parameters": {
            "sourceColumns": ["company"],
            "maskSymbol": "#",
            "customValue": "amazon"
        }
    }
}
```

MASK_DATE

Masker komponen tanggal dengan simbol topeng yang ditentukan pengguna.

Parameter

- `sourceColumns`— Daftar nama kolom yang ada.
- `maskSymbol`— Simbol yang akan digunakan untuk menggantikan karakter tertentu.
- `redact`— Sebuah array enum komponen tanggal untuk menutupi. Nilai enum yang valid: TAHUN, BULAN, HARI, JAM, MENIT, DETIK, MILIDETIK.
- `locale`- Tag bahasa IETF BCP 47 opsional. Default ke `en`. Lokal yang digunakan untuk pemformatan tanggal.

Example Contoh

```
// Mask year
{
    "RecipeAction": {
        "Operation": "MASK_DATE",
        "Parameters": {
            "sourceColumns": ["birthday"],
            "maskSymbol": "#",
            "redact": ["YEAR"]
        }
    }
}
```

TOPENG_PEMBATAS

Topeng karakter antara dua pembatas dengan simbol masking yang ditentukan pengguna.

Parameter

- `sourceColumns`— Daftar nama kolom yang ada.
- `maskSymbol`— Simbol yang akan digunakan untuk menggantikan karakter tertentu.
- `startDelimiter`— Karakter yang menunjukkan di mana masking harus dimulai. Menghilangkan parameter ini akan menerapkan topeng mulai dari awal string.
- `endDelimiter`— Karakter yang menunjukkan di mana masking akan berakhir. Menghilangkan parameter ini akan menerapkan masking dari startDelimiter ke akhir string.
- `preserveDelimiters`— Jika benar, terapkan masker ke pembatas.

- `alphabet`— Array set karakter untuk dilestarikan selama masking. Nilai enum yang valid: SIMBOL, SPASI.
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.

Example Contoh

```
// 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

Topeng karakter antara dua posisi dengan simbol masking yang ditentukan pengguna.

Parameter

- `sourceColumns`— Daftar nama kolom yang ada.
- `maskSymbol`— Simbol yang akan digunakan untuk menggantikan karakter tertentu.
- `start`— Angka yang menunjukkan posisi karakter mana masking akan dimulai (diindeks 0, inklusif). Pengindeksan negatif diperbolehkan. Menghilangkan parameter ini akan menerapkan topeng dari awal string hingga 'berhenti'.
- `stop`— Angka yang menunjukkan posisi karakter mana masking akan berakhir (diindeks 0, eksklusif). Pengindeksan negatif diperbolehkan. Menghilangkan parameter ini akan menerapkan mask dari 'start' sampai akhir string.

- `alphabet`— Array karakter menetapkan enum untuk diawetkan selama masking. Nilai enum yang valid: SIMBOL, SPASI.
- `entityTypeFilter`— Array opsional tipe entitas. Dapat digunakan untuk mengenkripsi hanya PII yang terdeteksi di kolom teks bebas.

Example Contoh

```
// Mask entire string
{
    "RecipeAction": {
        "Operation": "MASK_RANGE",
        "Parameters": {
            "sourceColumns": ["firstName", "lastName"],
            "maskSymbol": "#"
        }
    }
}
```

REPLACE_WITH_RANDOM_BETWEEN

Mengganti nilai dengan angka acak.

Parameter

- `lowerBound`— Batas bawah dari rentang angka acak.
- `sourceColumns`— Daftar nama kolom yang ada.
- `upperBound`— Batas atas rentang angka acak.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "REPLACE_WITH_RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "sourceColumns": ["column1", "column2"],
            "upperBound": "100"
        }
```

```
    }
}
```

REPLACE_WITH_RANDOM_DATE_BETWEEN

Mengganti nilai dengan tanggal acak.

Parameter

- `startDate`— Awal rentang tanggal dari mana tanggal acak akan diambil.
- `sourceColumns`— Daftar nama kolom yang ada.
- `endDate`— Akhir dari rentang tanggal dari mana tanggal acak akan diambil.

Example Contoh

```
{
    "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

Mengacak nilai dalam kolom tertentu. Pengocokan dapat terjadi dengan nilai yang dikelompokkan oleh kolom sekunder.

Parameter

- `sourceColumns`— Array kolom yang ada.
- `groupByColumns`— Array kolom untuk mengelompokkan kolom sumber dengan saat mengocoknya.

Example Contoh

```
{
   "sourceColumns": ["age"],
   "*groupByColumns*": ["country"]
}
```

Deteksi outlier dan langkah-langkah penanganan resep

Gunakan langkah-langkah resep ini untuk bekerja dengan outlier dalam data Anda dan melakukan transformasi lanjutan pada mereka..

Topik

- FLAG_OUTLIER
- REMOVE_OUTLIERS
- REPLACE_OUTLIERS
- RESCALE_OUTLIERS_WITH_Z_SCORE
- RESCALE_OUTLIERS_WITH_SKEW

FLAG_OUTLIER

Mengembalikan kolom baru yang berisi nilai disesuaikan di setiap baris yang menunjukkan jika nilai kolom sumber adalah outlier.

Parameter

- `sourceColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `targetColumn`— Menentukan nama kolom baru di mana hasil dari strategi evaluasi outlier akan dimasukkan.
- `outlierStrategy`— Menentukan pendekatan untuk digunakan dalam mendeteksi outlier. Nilai-nilai yang valid meliputi:
  - `Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi.
  - `MODIFIED_Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari median lebih dari ambang deviasi absolut median.
  - `IQR`— Mengidentifikasi nilai sebagai outlier ketika berada di luar kuartil pertama dan terakhir dari data kolom. Rentang interkuartil (IQR) mengukur di mana 50% titik data tengah berada.

- `threshold`- Menentukan nilai ambang untuk digunakan saat mendeteksi outlier. `sourceColumn`Nilai diidentifikasi sebagai outlier jika skor yang dihitung dengan `outlierStrategy` melebihi angka ini. Default-nya adalah 3.
- `trueString`- Menentukan nilai string untuk digunakan jika outlier terdeteksi. Defaultnya adalah “Benar”.
- `falseString`- Menentukan nilai string untuk digunakan jika tidak ada outlier terdeteksi. Defaultnya adalah “False”.

Contoh berikut menampilkan sintaks untuk satu RecipeActionoperasi. Resep berisi setidaknya satu RecipeStepoperasi, dan langkah resep berisi setidaknya satu tindakan resep. Tindakan resep menjalankan transformasi data yang Anda tentukan. Sekelompok tindakan resep berjalan secara berurutan untuk membuat kumpulan data akhir.

JSON

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Resep, menggunakan sintaks JSON. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di 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"
        }
    }
}
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

YAML

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Recipe, menggunakan sintaks YAMAL. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di YAMAL

```
- Action:
  Operation: FLAG_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: IQR
    trueString: Outlier
    falseString: No
    threshold: '1.5'
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

REMOVE_OUTLIERS

Menghapus titik data yang diklasifikasikan sebagai outlier, berdasarkan pengaturan dalam parameter.

Parameter

- `sourceColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `outlierStrategy`— Menentukan pendekatan untuk digunakan dalam mendeteksi outlier. Nilai-nilai yang valid meliputi:
  - `Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi.
  - `MODIFIED_Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari median lebih dari ambang deviasi absolut median.
  - `IQR`— Mengidentifikasi nilai sebagai outlier ketika berada di luar kuartil pertama dan terakhir dari data kolom. Rentang interkuartil (IQR) mengukur di mana 50% titik data tengah berada.

- `threshold`- Menentukan nilai ambang untuk digunakan saat mendeteksi outlier. `sourceColumn`Nilai diidentifikasi sebagai outlier jika skor yang dihitung dengan `outlierStrategy` melebihi angka ini. Default-nya adalah 3.
- `removeType`- Menentukan cara untuk menghapus data. Nilai yang valid mencakup `DELETE_ROWS` dan `CLEAR`.
- `trimValue`- Menentukan apakah akan menghapus semua atau beberapa outlier. Nilai Boolean ini default ke. `FALSE`
  - `FALSE`— Menghapus semua outlier
  - `TRUE`— Menghapus outlier yang berperingkat di luar ambang persentil yang ditentukan dalam dan. `minValue` `maxValue`
- `minValue`— Menunjukkan nilai persentil minimum untuk rentang outlier. Rentang yang valid adalah 0—100.
- `maxValue`— Menunjukkan nilai persentil maksimum untuk rentang outlier. Rentang yang valid adalah 0—100.

Contoh berikut menampilkan sintaks untuk satu RecipeActionoperasi. Resep berisi setidaknya satu RecipeStepoperasi, dan langkah resep berisi setidaknya satu tindakan resep. Tindakan resep menjalankan transformasi data yang Anda tentukan. Sekelompok tindakan resep berjalan secara berurutan untuk membuat kumpulan data akhir.

JSON

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Resep, menggunakan sintaks JSON. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di 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"
        }
    }
}
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

YAML

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Recipe, menggunakan sintaks YAMAL. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di YAMAL

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    outlierStrategy: Z_SCORE
    threshold: '3'
    removeType: DELETE_ROWS
    trimValue: 'TRUE'
    minValue: '5'
    maxValue: '95'
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

REPLACE_OUTLIERS

Memperbarui nilai titik data yang diklasifikasikan sebagai outlier, berdasarkan pengaturan dalam parameter.

Parameter

- `sourceColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.

- `outlierStrategy`— Menentukan pendekatan untuk digunakan dalam mendeteksi outlier. Nilai-nilai yang valid meliputi:
  - `Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi.
  - `MODIFIED_Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari median lebih dari ambang deviasi absolut median.
  - `IQR`— Mengidentifikasi nilai sebagai outlier ketika berada di luar kuartil pertama dan terakhir dari data kolom. Rentang interkuartil (IQR) mengukur di mana 50% titik data tengah berada.
- `threshold`- Menentukan nilai ambang untuk digunakan saat mendeteksi outlier. `sourceColumn`Nilai diidentifikasi sebagai outlier jika skor yang dihitung dengan `outlierStrategy` melebihi angka ini. Default-nya adalah 3.
- `replaceType`- Menentukan metode yang akan digunakan saat mengganti outlier. Nilai-nilai yang valid meliputi:
  - `WINSORIZE_VALUES`- Menentukan menggunakan persentil minimum dan maksimum untuk membatasi nilai-nilai.
  - `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`— Menunjukkan jenis fungsi modal yang akan `replaceType` digunakan kapan`REPLACE_WITH_MODE`. Nilai yang valid meliputi:`MIN`,`MAX`, dan`AVERAGE`.
- `minValue`— Menunjukkan nilai persentil minimum untuk rentang outlier yang akan diterapkan saat digunakan. `trimValue` Rentang yang valid adalah 0—100.
- `maxValue`— Menunjukkan nilai persentil maksimum untuk rentang outlier yang akan diterapkan saat `trimValue` digunakan. Rentang yang valid adalah 0—100.
- `value`- Menentukan nilai untuk memasukkan saat menggunakan`REPLACE_WITH_CUSTOM`.

- `trimValue`- Menentukan apakah akan menghapus semua atau beberapa outlier. Nilai Boolean ini diatur ke `TRUE` when `replaceType` is`REPLACE_WITH_NULL`,`REPLACE_WITH_MODE`, or`WINSORIZE_VALUES`. Ini default untuk `FALSE` semua yang lain.
  - `FALSE`— Menghapus semua outlier
  - `TRUE`—Menghapus outlier yang berperingkat di luar ambang batas persentil yang ditentukan dalam dan. `minValue` `maxValue`

Contoh berikut menampilkan sintaks untuk satu RecipeActionoperasi. Resep berisi setidaknya satu RecipeStepoperasi, dan langkah resep berisi setidaknya satu tindakan resep. Tindakan resep menjalankan transformasi data yang Anda tentukan. Sekelompok tindakan resep berjalan secara berurutan untuk membuat kumpulan data akhir.

JSON

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Resep, menggunakan sintaks JSON. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di 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"
        }
    }
}
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

YAML

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Recipe, menggunakan sintaks YAMAL. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di YAMAL

```
- 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'
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

RESCALE_OUTLIERS_WITH_Z_SCORE

Mengembalikan kolom baru dengan nilai outlier rescaled di setiap baris, berdasarkan pengaturan dalam parameter. Tindakan ini juga menerapkan Z-score normalisasi pada nilai data skala linier untuk memiliki rata-rata (μ) 0 dan standar deviasi (σ) 1. Kami merekomendasikan tindakan ini untuk menangani outlier.

Parameter

- `sourceColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `targetColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `outlierStrategy`— Menentukan pendekatan untuk digunakan dalam mendeteksi outlier. Nilai-nilai yang valid meliputi:
  - `Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi.

  - `MODIFIED_Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari median lebih dari ambang deviasi absolut median.
  - `IQR`— Mengidentifikasi nilai sebagai outlier ketika berada di luar kuartil pertama dan terakhir dari data kolom. Rentang interkuartil (IQR) mengukur di mana 50% titik data tengah berada.
- `threshold`— Nilai ambang batas yang digunakan saat mendeteksi outlier. `sourceColumn`Nilai diidentifikasi sebagai outlier jika skor yang dihitung dengan `outlierStrategy` melebihi angka ini. Default-nya adalah 3.

Contoh berikut menampilkan sintaks untuk satu RecipeActionoperasi. Resep berisi setidaknya satu RecipeStepoperasi, dan langkah resep berisi setidaknya satu tindakan resep. Tindakan resep menjalankan transformasi data yang Anda tentukan. Sekelompok tindakan resep berjalan secara berurutan untuk membuat kumpulan data akhir.

JSON

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk operasi DataBrew Resep, menggunakan sintaks JSON. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di JSON

```
{
    "Action": {
        "Operation": "RESCALE_OUTLIERS_WITH_Z_SCORE",
        "Parameters": {
            "sourceColumn": "name-of-existing-column",
            "targetColumn": "name-of-new-column",
            "outlierStrategy": "Z_SCORE",
            "threshold": "3"
        }
    }
}
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

YAML

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk operasi DataBrew Recipe, menggunakan sintaks YAMAL. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di YAMAL

```
- Action:
  Operation: REMOVE_OUTLIERS
  Parameters:
    sourceColumn: name-of-existing-column
    targetColumn: name-of-new-column
    outlierStrategy: Z_SCORE
    threshold: '3'
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

RESCALE_OUTLIERS_WITH_SKEW

Mengembalikan kolom baru dengan nilai outlier rescaled di setiap baris, berdasarkan pengaturan dalam parameter. Tindakan ini berfungsi untuk mengurangi kemiringan distribusi dengan menerapkan log atau transformasi root yang ditentukan. Kami merekomendasikan tindakan ini untuk menangani data miring.

Parameter

- `sourceColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `targetColumn`- Menentukan nama kolom numerik yang ada yang mungkin berisi outlier.
- `outlierStrategy`— Menentukan pendekatan untuk digunakan dalam mendeteksi outlier. Nilai-nilai yang valid meliputi:
  - `Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari rata-rata dengan lebih dari ambang standar deviasi.
  - `MODIFIED_Z_SCORE`— Mengidentifikasi nilai sebagai outlier ketika menyimpang dari median lebih dari ambang deviasi absolut median.

  - `IQR`— Mengidentifikasi nilai sebagai outlier ketika berada di luar kuartil pertama dan terakhir dari data kolom. Rentang interkuartil (IQR) mengukur di mana 50% titik data tengah berada.
- `threshold`- Menentukan nilai ambang untuk digunakan saat mendeteksi outlier. `sourceColumn`Nilai diidentifikasi sebagai outlier jika skor yang dihitung dengan `outlierStrategy` melebihi angka ini. Default-nya adalah 3.
- `skewFunction`- Menentukan metode yang akan digunakan saat mengganti outlier. Nilai-nilai yang valid meliputi:
  - LOG — Menerapkan transformasi yang kuat untuk mengurangi kemiringan positif dan negatif. Ini adalah logaritma natural (2.718281828).
  - ROOT (with`value = 3`) — Menerapkan transformasi yang cukup kuat untuk mengurangi kemiringan positif dan negatif. (Akar kubus)
  - ROOT (with`value = 2`) — Menerapkan transformasi moderat untuk mengurangi kemiringan positif saja. (Akar kuadrat)
  - SQUARE — Menerapkan transformasi moderat untuk mengurangi kemiringan negatif. (Persegi)
  - Transformasi kustom - Menerapkan yang ditentukan `LOG` atau `ROOT` mengubah menggunakan nomor kustom yang disediakan dalam `value` parameter.
- `value`- Menentukan nilai yang akan digunakan untuk transformasi kustom. Jika `skewFunction` LOG, nilai ini mewakili dasar log. Jika `skewFunction` ROOT, nilai ini mewakili kekuatan root.

Contoh berikut menampilkan sintaks untuk satu RecipeActionoperasi. Resep berisi setidaknya satu RecipeStepoperasi, dan langkah resep berisi setidaknya satu tindakan resep. Tindakan resep menjalankan transformasi data yang Anda tentukan. Sekelompok tindakan resep berjalan secara berurutan untuk membuat kumpulan data akhir.

JSON

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Resep, menggunakan sintaks JSON. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di 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"
        }
    }
}
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

YAML

Berikut ini menunjukkan contoh `RecipeAction` untuk digunakan sebagai anggota contoh `RecipeStep` untuk DataBrew Recipe, menggunakan sintaks YAMAL. Untuk contoh sintaks yang menampilkan daftar tindakan resep, lihatMendefinisikan struktur resep.

Example Contoh di YAMAL

```
- 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'
```

Untuk informasi selengkapnya tentang penggunaan tindakan resep ini dalam operasi API, lihat CreateRecipeatau UpdateRecipe. Anda dapat menggunakan ini dan operasi API lainnya dalam kode Anda sendiri.

Langkah-langkah resep struktur kolom

Gunakan langkah-langkah resep struktur kolom ini untuk memodifikasi struktur kolom data Anda.

Topik

- BOOLEAN_OPERASI
- 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_INTERVAL

BOOLEAN_OPERASI

Buat kolom baru, berdasarkan hasil kondisi logis IF. Mengembalikan nilai true jika ekspresi boolean adalah true, nilai false jika ekspresi boolean adalah false, atau mengembalikan nilai kustom.

Parameter

- `trueValueExpression`— Hasil ketika kondisi terpenuhi.
- `falseValueExpression`— Hasil ketika kondisi tidak terpenuhi.
- `valueExpression`— Kondisi Boolean.
- `withExpressions`— Konfigurasi untuk hasil agregat.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Anda dapat menggunakan nilai konstan, referensi kolom, dan hasil agregat dalam trueValueExpression, false ValueExpression dan valueExpression.

Example Contoh: Nilai konstan

Nilai yang tetap tidak berubah, seperti angka atau kalimat.

```
{
  "RecipeStep": {
```

```
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Contoh: Referensi kolom

Nilai yang merupakan kolom dalam kumpulan data.

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "`column.2`",
        "falseValueExpression": "`column.3`",
        "valueExpression": "`column.1` < `column.4`",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Contoh: Hasil agregat

Nilai yang dihitung oleh fungsi agregat. Fungsi agregat melakukan perhitungan pada kolom, dan mengembalikan nilai tunggal.

```
{
  "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"
      }
    }
  }
}
```

Pengguna perlu mengonversi JSON ke string dengan melarikan diri.

Perhatikan bahwa nama parameter dalam trueValueExpression, falseValueExpression, dan valueExpression harus cocok dengan nama di WithExpressions. Untuk menggunakan hasil agregat dari beberapa kolom, Anda perlu membuat parameter untuk mereka dan menyediakan fungsi agregat.

Example Contoh:

```
{
  "RecipeStep": {
    "Action": {
      "Operation": "BOOLEAN_OPERATION",
      "Parameters": {
        "trueValueExpression": "It is true.",
        "falseValueExpression": "It is false.",
        "valueExpression": "`column.1` < 2000",
        "targetColumn": "result.column"
      }
    }
  }
}
```

Example Contoh: and/or

Anda dapat menggunakan dan dan atau untuk menggabungkan beberapa kondisi.

```
{
  "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"
      }
    }
  }
}
```

Fungsi agregat yang valid

Tabel di bawah ini menunjukkan semua fungsi agregat valid yang dapat digunakan dalam operasi boolean.

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
Numerik	Jumlah	`:sum.column.1`	`[ { "name": "sum.column.1", "value":`	Mengembalikan jumlah `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan

			<pre>"sum(`column.1`)",
			
			"type": "aggregate"
			}
			]</pre>	

	Berarti	`:maksud.column.1`	<pre>[
			{
			
			"name": "mean.column.1",
			
			"value": "avg(`column.1`)",
			
			"type": "aggregate"
			}
			]</pre>
				Mengembalikan mean dari `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Berarti deviasi absolut	`:meanabsolutedeviation.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Mengembalikan deviasi absolut rata-rata `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Median	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Mengembalikan median `column.1`
	Produk	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Mengembalikan produk dari `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Standar deviasi	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Mengembalikan standar deviasi `column.1`
	Varians	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Mengembalikan varians `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Kesalahan standar rata-rata	`:standarderrorofmean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kesalahan standar rata-rata `column.1`
	Kemiringan	`:skewness.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kemiringan `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kurtosis `column.1`
Datetime/ Numeric/Text	Hitungan	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Mengembalikan jumlah total baris di `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Hitung berbeda	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Mengembalikan jumlah total baris yang berbeda di `column.1`
	Min	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Mengembalikan nilai minimum `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Maks	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Mengembalikan nilai maksimum `column.1`

Kondisi yang valid dalam ValueExpression

Tabel di bawah ini menunjukkan kondisi yang didukung dan ekspresi nilai yang dapat Anda gunakan.

Jenis kolom	Kondisi	ValueExpression	Deskripsi
String	Contains	berisi (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom berisi teks
	Tidak mengandung	! berisi (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak mengandung teks
	Cocok	cocok (`kolom`, 'pola')	Kondisi untuk menguji apakah nilai dalam

Jenis kolom	Kondisi	ValueExpression	Deskripsi
			kolom cocok dengan pola
	Tidak cocok	! cocok (`kolom`, 'pola')	Kondisi untuk menguji apakah nilai dalam kolom tidak cocok dengan pola
	Starts with	startsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom dimulai dengan teks
	Tidak dimulai dengan	! startsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak dimulai dengan teks
	Ends with	EndsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom berakhir dengan teks
	Tidak berakhir dengan	! EndsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak diakhiri dengan teks
Numerik	Kurang dari	`kolom` < nomor	Kondisi untuk menguji apakah nilai dalam kolom kurang dari angka

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Kurang dari atau sama dengan	`kolom` <= angka	Kondisi untuk menguji apakah nilai dalam kolom kurang dari atau sama dengan angka
	Lebih besar dari	`kolom` > nomor	Kondisi untuk menguji apakah nilai dalam kolom lebih besar dari angka
	Lebih besar dari atau sama dengan	`kolom` >= nomor	Kondisi untuk menguji apakah nilai dalam kolom lebih besar dari atau sama dengan angka
	Adalah antara	isBetween (`kolom`, minNumber, maxNumber)	Kondisi untuk menguji apakah nilai dalam kolom berada di antara minNumber dan maxNumber
	Bukan di antara	! isBetween (`kolom`, minNumber, maxNumber)	Kondisi untuk menguji apakah nilai dalam kolom tidak berada di antara minNumber dan maxNumber
Boolean	Apakah benar	`kolom` = BENAR	Kondisi untuk menguji apakah nilai dalam kolom boolean TRUE
	Adalah salah	`kolom` = SALAH	Kondisi untuk menguji apakah nilai dalam kolom boolean FALSE

Jenis kolom	Kondisi	ValueExpression	Deskripsi
Date/Timestamp	Lebih awal dari	`kolom` < 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih awal dari tanggal
	Lebih awal dari atau sama dengan	`kolom` <= 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih awal dari atau sama dengan tanggal
	Lebih lambat dari	`kolom` > 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih lambat dari tanggal
	Lebih lambat dari atau sama dengan	`kolom` >= 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih lambat dari atau sama dengan tanggal
String/Numeric/Date/ Timestamp	Tepat	`kolom` = 'nilai'	Kondisi untuk menguji apakah nilai dalam kolom persis nilai
	Is not	`kolom`! = 'nilai'	Kondisi untuk menguji apakah nilai dalam kolom bukan nilai
	Hilang	isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom hilang

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Tidak hilang	! isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom tidak hilang
	Valid	isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)
	Tidak valid	! isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom tidak valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)
Bersarang	Hilang	isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom hilang
	Tidak hilang	! isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom tidak hilang
	Valid	isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Tidak valid	! isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom tidak valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)

CASE_OPERATION

Buat kolom baru, berdasarkan hasil dari kondisi logis CASE. Operasi kasus melewati kondisi kasus dan mengembalikan nilai ketika kondisi pertama terpenuhi. Setelah kondisi benar, operasi berhenti membaca dan mengembalikan hasilnya. Jika tidak ada kondisi yang benar, ia mengembalikan nilai default.

Parameter

- `valueExpression`— Kondisi.
- `withExpressions`— Konfigurasi untuk hasil agregat.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
  "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"
      }
    }
  }
}
```

Fungsi agregat yang valid

Tabel di bawah ini menunjukkan semua fungsi agregat yang valid yang dapat digunakan dalam operasi kasus.

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
Numerik	Jumlah	`:sum.column.1`	`[ { "name": "sum.column.1", "value": "sum(`column.1`)", "type": "aggregate" } ]`	Mengembalikan jumlah `column.1`
	Berarti	`:maksud.column.1`	`[ { "name": "mean.column.1", "value": "avg(`column.1`)", "type": "aggregate" }`	Mengembalikan rata-rata `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
			`]`	
	Berarti deviasi absolut	`:meanabsolutedeviation.column.1`	`[ { "name": "meanabsolutedeviation.column.1", "value": "mean_absolute_deviation(`column.1`)", "type": "aggregate" } ]`	Mengembalikan deviasi absolut rata-rata `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Median	`:median.column.1`	`[ { "name": "median.column.1", "value": "median(`column.1`)", "type": "aggregate" } ]`	Mengembalikan median `column.1`
	Produk	`:product.column.1`	`[ { "name": "product.column.1", "value": "product(`column.1`)", "type": "aggregate" } ]`	Mengembalikan produk dari `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Standar deviasi	`:standarddeviation.column.1`	`[ { "name": "standarddeviation.column.1", "value": "stddev(`column.1`)", "type": "aggregate" } ]`	Mengembalikan standar deviasi `column.1`
	Varians	`:variance.column.1`	`[ { "name": "variance.column.1", "value": "variance(`column.1`)", "type": "aggregate" } ]`	Mengembalikan varians `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Kesalahan standar rata-rata	`:standarderrorofmean.column.1`	`[ { "name": "standarderrorofmean.column.1", "value": "standard_error_of_mean(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kesalahan standar rata-rata `column.1`
	Kemiringan	`:kemiringan.column.1`	`[ { "name": "skewness.column.1", "value": "skewness(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kemiringan `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Kurtosis	`:kurtosis.column.1`	`[ { "name": "kurtosis.column.1", "value": "kurtosis(`column.1`)", "type": "aggregate" } ]`	Mengembalikan kurtosis `column.1`
Datetime/ Numeric/Text	Hitungan	`:count.column.1`	`[ { "name": "count.column.1", "value": "count(`column.1`)", "type": "aggregate" } ]`	Mengembalikan jumlah total baris di `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Hitung berbeda	`:countdistinct.column.1`	`[ { "name": "count.column.1", "value": "count(distinct `column.1`)", "type": "aggregate" } ]`	Mengembalikan jumlah total baris yang berbeda di `column.1`
	Min	`:min.column.1`	`[ { "name": "min.column.1", "value": "min(`column.1`)", "type": "aggregate" } ]`	Mengembalikan nilai minimum `column.1`

Jenis kolom	Kondisi	ValueExpression	denganXpressions	Nilai yang dikembalikan
	Maks	`:max.column.1`	`[ { "name": "max.column.1", "value": "max(`column.1`)", "type": "aggregate" } ]`	Mengembalikan nilai maksimum `column.1`

Kondisi yang valid dalam ValueExpression

Tabel di bawah ini menunjukkan kondisi yang didukung dan ekspresi nilai yang dapat Anda gunakan.

Jenis kolom	Kondisi	ValueExpression	Deskripsi
String	Contains	berisi (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom berisi teks
	Tidak mengandung	! berisi (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak mengandung teks
	Cocok	cocok (`kolom`, 'pola')	Kondisi untuk menguji apakah nilai dalam

Jenis kolom	Kondisi	ValueExpression	Deskripsi
			kolom cocok dengan pola
	Tidak cocok	! cocok (`kolom`, 'pola')	Kondisi untuk menguji apakah nilai dalam kolom tidak cocok dengan pola
	Starts with	startsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom dimulai dengan teks
	Tidak dimulai dengan	! startsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak dimulai dengan teks
	Ends with	EndsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom berakhir dengan teks
	Tidak berakhir dengan	! EndsWith (`kolom`, 'teks')	Kondisi untuk menguji apakah nilai dalam kolom tidak diakhiri dengan teks
Numerik	Kurang dari	`kolom` < nomor	Kondisi untuk menguji apakah nilai dalam kolom kurang dari angka

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Kurang dari atau sama dengan	`kolom` <= angka	Kondisi untuk menguji apakah nilai dalam kolom kurang dari atau sama dengan angka
	Lebih besar dari	`kolom` > nomor	Kondisi untuk menguji apakah nilai dalam kolom lebih besar dari angka
	Lebih besar dari atau sama dengan	`kolom` >= nomor	Kondisi untuk menguji apakah nilai dalam kolom lebih besar dari atau sama dengan angka
	Adalah antara	isBetween (`kolom`, minNumber, maxNumber)	Kondisi untuk menguji apakah nilai dalam kolom berada di antara minNumber dan maxNumber
	Bukan di antara	! isBetween (`kolom`, minNumber, maxNumber)	Kondisi untuk menguji apakah nilai dalam kolom tidak berada di antara minNumber dan maxNumber
Boolean	Apakah benar	`kolom` = BENAR	Kondisi untuk menguji apakah nilai dalam kolom boolean TRUE
	Adalah salah	`kolom` = SALAH	Kondisi untuk menguji apakah nilai dalam kolom boolean FALSE

Jenis kolom	Kondisi	ValueExpression	Deskripsi
Date/Timestamp	Lebih awal dari	`kolom` < 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih awal dari tanggal
	Lebih awal dari atau sama dengan	`kolom` <= 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih awal dari atau sama dengan tanggal
	Lebih lambat dari	`kolom` > 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih lambat dari tanggal
	Lebih lambat dari atau sama dengan	`kolom` >= 'tanggal'	Kondisi untuk menguji apakah nilai dalam kolom lebih lambat dari atau sama dengan tanggal
String/Numeric/Date/ Timestamp	Tepat	`kolom` = 'nilai'	Kondisi untuk menguji apakah nilai dalam kolom persis nilai
	Is not	`kolom`! = 'nilai'	Kondisi untuk menguji apakah nilai dalam kolom bukan nilai
	Hilang	isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom hilang

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Tidak hilang	! isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom tidak hilang
	Valid	isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)
	Tidak valid	! isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom tidak valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)
Bersarang	Hilang	isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom hilang
	Tidak hilang	! isMissing (`kolom`)	Kondisi untuk menguji apakah nilai dalam kolom tidak hilang
	Valid	isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)

Jenis kolom	Kondisi	ValueExpression	Deskripsi
	Tidak valid	! isValid (`kolom`, tipe data)	Kondisi untuk menguji apakah nilai dalam kolom tidak valid (nilainya adalah tipe data atau dapat dikonversi ke tipe data)

FLAG_COLUMN_FROM_NULL

Membuat kolom baru, berdasarkan keberadaan nilai null di kolom yang ada.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `flagType`— Nilai yang harus diatur ke`Null values`.
- `trueString`— Nilai untuk kolom baru, jika nilai null ditemukan di sumber. Jika tidak ada nilai yang ditentukan, default-nya adalah `True`.
- `falseString`— Nilai untuk kolom baru, jika nilai non-null ditemukan di sumber. Jika tidak ada nilai yang ditentukan, default-nya adalah `False`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_NULL",
        "Parameters": {
            "flagType": "Null values",
            "sourceColumn": "weight_kg",
            "targetColumn": "is_weight_kg_missing"
        }
    }
}
```

FLAG_COLUMN_FROM_PATTERN

Membuat kolom baru, berdasarkan keberadaan pola yang ditentukan pengguna di kolom yang ada.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `flagType`— Nilai yang harus diatur ke`Pattern`.
- `pattern`— Ekspresi reguler, menunjukkan pola yang akan dievaluasi.
- `trueString`— Nilai untuk kolom baru, jika nilai null ditemukan di sumber. Jika tidak ada nilai yang ditentukan, default-nya adalah `True`.
- `falseString`— Nilai untuk kolom baru, jika nilai non-null ditemukan di sumber. Jika tidak ada nilai yang ditentukan, default-nya adalah `False`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FLAG_COLUMN_FROM_PATTERN",
        "Parameters": {
            "falseString": "No",
            "flagType": "Pattern",
            "pattern": "N.*",
            "sourceColumn": "wind_direction",
            "targetColumn": "northerly",
            "trueString": "yes"
        }
    }
}
```

MERGE

Menggabungkan dua atau lebih kolom ke dalam kolom baru.

Parameter

- `sourceColumns`— JSON-encoded String yang mewakili daftar satu atau lebih kolom yang akan digabungkan.

- `delimiter`— Pemisah opsional antara nilai-nilai, untuk muncul di kolom target.
- `targetColumn`— Nama kolom gabungan yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MERGE",
        "Parameters": {
            "delimiter": " ",
            "sourceColumns": "[\"first_name\",\"last_name\"]",
            "targetColumn": "Merged Column 1"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_DELIMITER

Membagi kolom menjadi tiga kolom baru, sesuai dengan pembatas awal dan akhir.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `patternOption1`— JSON-encoded String yang mewakili satu atau lebih karakter yang menunjukkan pembatas pertama.
- `patternOption2`— JSON-encoded String yang mewakili satu atau lebih karakter yang menunjukkan pembatas kedua.
- `pattern`— Satu atau lebih karakter untuk digunakan sebagai pemisah, saat membagi data.
- `includeInSplit`— Jika benar, sertakan pola di kolom baru; jika tidak, polanya dibuang.

Example Contoh

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

```
            "patternOption1": "{\"pattern\":\"H\",\"includeInSplit\":true}",
            "patternOption2": "{\"pattern\":\"M\",\"includeInSplit\":true}",
            "sourceColumn": "last_name"
        }
    }
}
```

SPLIT_COLUMN_BETWEEN_POSITIONS

Membagi kolom menjadi tiga kolom baru, sesuai dengan offset yang Anda tentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `startPosition`— Posisi karakter di mana perpecahan akan dimulai.
- `endPosition`— Posisi karakter di mana perpecahan berakhir.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_BETWEEN_POSITIONS",
        "Parameters": {
            "endPosition": "12",
            "sourceColumn": "last_name",
            "startPosition": "2"
        }
    }
}
```

SPLIT_COLUMN_FROM_END

Membagi kolom menjadi dua kolom baru, pada offset dari akhir string.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `position`— Posisi karakter, dari ujung kanan string, di mana perpecahan akan terjadi.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_END",
        "Parameters": {
            "position": "1",
            "sourceColumn": "nationality"
        }
    }
}
```

SPLIT_COLUMN_FROM_START

Membagi kolom menjadi dua kolom baru, pada offset dari awal string.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `position`— Posisi karakter, dari ujung kiri string, di mana perpecahan akan terjadi.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_FROM_START",
        "Parameters": {
            "position": "1",
            "sourceColumn": "first_name"
        }
    }
}
```

SPLIT_COLUMN_MULTIPLE_DELIMITER

Membagi kolom menurut beberapa pembatas.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `patternOptions`— JSON-encoded String yang mewakili satu atau lebih pola yang menentukan kriteria split.
- `pattern`— Satu atau lebih karakter untuk digunakan sebagai pemisah, saat membagi data.
- `limit`— Berapa banyak split yang harus dilakukan. Minimal adalah 1; maksimum adalah 20.
- `includeInSplit`— Jika benar, sertakan pola di kolom baru; jika tidak, polanya dibuang.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_MULTIPLE_DELIMITER",
        "Parameters": {
            "limit": "1",
            "patternOptions": "[{\"pattern\":\",\",\"includeInSplit\":true},{\"pattern
\":\" \",\"includeInSplit\":true}]",
            "sourceColumn": "description"
        }
    }
}
```

SPLIT_COLUMN_SINGLE_DELIMITER

Membagi kolom menjadi satu atau lebih kolom baru, sesuai dengan pembatas tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `pattern`— Satu atau lebih karakter untuk digunakan sebagai pemisah, saat membagi data.
- `limit`— Berapa banyak split yang harus dilakukan. Minimal adalah 1; maksimum adalah 20.
- `includeInSplit`— Jika benar, sertakan pola di kolom baru; jika tidak, polanya dibuang.

Example Contoh

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

```
            "includeInSplit": "true",
            "limit": "1",
            "pattern": "/",
            "sourceColumn": "info_url"
        }
    }
}
```

SPLIT_COLUMN_WITH_INTERVAL

Membagi kolom pada interval n karakter, di mana Anda menentukan n.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `startPosition`— Posisi karakter di mana perpecahan akan dimulai.
- `interval`— Jumlah karakter yang harus dilewati sebelum perpecahan berikutnya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SPLIT_COLUMN_WITH_INTERVALS",
        "Parameters": {
            "interval": "4",
            "sourceColumn": "nationality",
            "startPosition": "1"
        }
    }
}
```

Langkah resep pemformatan kolom

Gunakan langkah-langkah resep pemformatan kolom untuk mengubah format data di kolom Anda.

Topik

- NOMOR_FORMAT
- FORMAT_PHONE_NUMBER

NOMOR_FORMAT

Mengembalikan kolom di mana nilai numerik diubah menjadi string diformat.

Parameter

- `sourceColumn` – String. Nama kolom yang ada.
- `decimalPlaces`— Bilangan bulat. Nilai jumlah digit setelah pemisah desimal.
- `numericDecimalSeparator` – String. Salah satu nilai berikut yang menunjukkan pemisah desimal:
  - "."
  - ","
- `numericThousandSeparator` – String. Salah satu nilai berikut yang menunjukkan pemisah seribu:
  - nol. Menunjukkan bahwa seribu pemisah tidak diaktifkan.
  - ","
  - " "
  - "."
  - "\\"
- `numericAbbreviatedUnit` – String. Salah satu nilai berikut yang menunjukkan unit singkatan:
  - nol. Menunjukkan bahwa unit singkatan tidak diaktifkan.
  - “SERIBU”
  - “JUTA”
  - “MILIAR”
  - “TRILIUN”
- `numericUnitAbbreviation` – String. Salah satu nilai berikut atau nilai kustom apa pun, menunjukkan singkatan unit:
  - nol. Menunjukkan bahwa singkatan unit tidak diaktifkan.
  -

Satuan singkatan	Opsi
Ribuan	K, k, M, seribu, adat
Juta	M, m, MM, juta, kustom

Satuan singkatan	Opsi
Miliar	B, bn, miliar, adat
Triliun	T, tn, triliun, kebiasaan

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NUMBER_FORMAT",
        "Parameters": {
            "sourceColumn": "income",
            "decimalPlaces": "2",
            "numericDecimalSeparator": ".",
            "numericThousandSeparator": ",",
            "numericAbbreviatedUnit": "THOUSAND",
            "numericUnitAbbreviation": "K"
        }
    }
}
```

FORMAT_PHONE_NUMBER

Mengembalikan kolom di mana string nomor telepon diubah menjadi nilai diformat.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `phoneNumberFormat`— Format untuk mengonversi nomor telepon menjadi. Jika tidak ada format yang ditentukan, defaultnya adalah`E.164`, format nomor telepon standar yang diakui secara internasional. Nilai-nilai yang valid meliputi:
  - `E164`(hilangkan periode setelahnya`E`)
- `defaultRegion`— Kode wilayah yang valid yang terdiri dari dua atau tiga huruf besar yang menentukan wilayah untuk nomor telepon ketika tidak ada kode negara yang ada di nomor itu sendiri. Paling-paling, salah satu `defaultRegion` atau `defaultRegionColumn` dapat disediakan.

- `defaultRegionColumn`— Nama kolom tipe data lanjutan`Country`. Kode wilayah dari kolom yang ditentukan digunakan untuk menentukan kode negara untuk nomor telepon ketika tidak ada kode negara dalam nomor itu sendiri. Paling-paling, salah satu `defaultRegion` atau `defaultRegionColumn` dapat disediakan.

Catatan

- Masukan yang tidak dapat diformat ke nomor telepon yang valid tetap tidak dimodifikasi.
- Jika tidak ada wilayah default yang disediakan, dan nomor telepon tidak dimulai dengan simbol plus (+) dan kode panggilan negara, nomor telepon tidak diformat.

Example

Contoh: Wilayah default Tetap

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegion": "US"
        }
    }
}
```

Contoh: Opsi kolom wilayah default

```
{
    "Action": {
        "Operation": "FORMAT_PHONE_NUMBER",
        "Parameters": {
            "sourceColumn": "Phone Number",
            "defaultRegionColumn": "Country Code"
        }
    }
}
```

Langkah-langkah resep struktur data

Gunakan langkah-langkah resep ini untuk mentabulasi dan meringkas data dari perspektif yang berbeda, atau untuk melakukan fungsi lanjutan.

Topik

- NEST_TO_ARRAY
- NEST_TO_MAP
- NEST_TO_STRUCT
- UNNEST_ARRAY
- UNNEST_MAP
- UNNEST_STRUCT
- UNNEST_STRUCT_N
- GROUP_BY
- BERGABUNG
- POROS
- SKALA
- MENTRANSPOS
- UNION
- UNPIVOT

NEST_TO_ARRAY

Mengkonversi kolom yang dipilih pengguna menjadi nilai array. Urutan kolom yang dipilih dipertahankan saat membuat array yang dihasilkan. Tipe data kolom yang berbeda adalah typecast ke tipe umum yang mendukung tipe data dari semua kolom.

Parameter

- `sourceColumns`— Daftar kolom sumber.
- `targetColumn`— Nama kolom target.
- `removeSourceColumns`— Berisi nilai `true` atau `false` untuk menunjukkan apakah pengguna ingin menghapus kolom sumber yang dipilih atau tidak.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_ARRAY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

NEST_TO_MAP

Mengkonversi kolom yang dipilih pengguna menjadi pasangan kunci-nilai, masing-masing dengan kunci yang mewakili nama kolom dan nilai yang mewakili nilai baris. Urutan kolom yang dipilih tidak dipertahankan saat membuat peta yang dihasilkan. Tipe data kolom yang berbeda adalah typecast ke tipe umum yang mendukung tipe data dari semua kolom.

Parameter

- `sourceColumns`— Daftar kolom sumber.
- `targetColumn`— Nama kolom target.
- `removeSourceColumns`— Berisi nilai `true` atau `false` untuk menunjukkan apakah pengguna ingin menghapus kolom sumber yang dipilih atau tidak.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_MAP",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
```

```
}
```

NEST_TO_STRUCT

Mengkonversi kolom yang dipilih pengguna menjadi pasangan kunci-nilai, masing-masing dengan kunci yang mewakili nama kolom dan nilai yang mewakili nilai baris. Urutan kolom yang dipilih dan tipe data dari setiap kolom dipertahankan dalam struct yang dihasilkan.

Parameter

- `sourceColumns`— Daftar kolom sumber.
- `targetColumn`— Nama kolom target.
- `removeSourceColumns`— Berisi nilai `true` atau `false` untuk menunjukkan apakah pengguna ingin menghapus kolom sumber yang dipilih atau tidak.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NEST_TO_STRUCT",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "columnName",
            "removeSourceColumns": "true"
        }
    }
}
```

UNNEST_ARRAY

Melepaskan kolom tipe `array` ke kolom baru. Jika array berisi lebih dari satu nilai, maka baris yang sesuai dengan setiap elemen dihasilkan. Fungsi ini hanya menghapus satu tingkat kolom array.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom ini harus `struct` bertipe.
- `targetColumn`— Nama kolom target yang dihasilkan.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNNEST_ARRAY",
        "Parameters": {
            "sourceColumn": "address",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_MAP

Melepaskan kolom tipe `map` dan menghasilkan kolom untuk kunci dan nilai. Jika ada lebih dari satu pasangan kunci-nilai, baris yang sesuai dengan setiap nilai kunci akan dihasilkan. Fungsi ini hanya menghapus satu tingkat kolom peta.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom ini harus `struct` bertipe.
- `removeSourceColumn`— Jika`true`, kolom sumber dihapus setelah fungsi selesai.
- `targetColumn`— Jika disediakan, masing-masing kolom yang dihasilkan akan dimulai dengan ini sebagai awalan.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNNEST_MAP",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false",
            "targetColumn": "address"
        }
    }
}
```

UNNEST_STRUCT

Unnest kolom tipe `struct` dan menghasilkan kolom untuk masing-masing kunci yang ada di struct. Fungsi ini hanya menghapus tingkat struct satu.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom ini harus dari tipe struct.
- `removeSourceColumn`— Jika`true`, kolom sumber dihapus setelah fungsi selesai.
- `targetColumn`— Jika disediakan, masing-masing kolom yang dihasilkan akan dimulai dengan ini sebagai awalan.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNNEST_STRUCT",
        "Parameters": {
            "sourceColumn": "address",
            "removeSourceColumn": "false"
            "targetColumn": "add"
        }
    }
}
```

UNNEST_STRUCT_N

Membuat kolom baru untuk setiap bidang kolom jenis yang dipilih`struct`.

Misalnya, mengingat struct berikut:

```
           user {
              name: “Ammy”
              address: {
                 state: "CA",
                 zipcode: 12345
              }
           }
```

Fungsi ini menciptakan 3 kolom:

user.name	user.address.state	user.address.zipcode
Ammy	CA	12345

Parameter

- `sourceColumns`— Daftar kolom sumber.
- `regexColumnSelector`— Ekspresi reguler untuk memilih kolom untuk unnest.
- `removeSourceColumn`— Nilai Boolean. Jika benar, maka hapus kolom sumber; jika tidak, simpan.
- `unnestLevel`— Jumlah level untuk unnest.
- `delimiter`— Pembatas digunakan dalam nama kolom yang baru dibuat untuk memisahkan tingkat yang berbeda dari struct. Misalnya: jika pembatas adalah “/”, nama kolom akan berada dalam bentuk ini: “user/address/state”.
- `conditionExpressions`— Ekspresi kondisi.

Example Contoh

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

GROUP_BY

Merangkum data dengan mengelompokkan baris dengan satu atau lebih kolom, dan kemudian menerapkan fungsi agregasi untuk setiap grup.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang membentuk dasar dari setiap kelompok.
- `groupByAggFunctions`— Sebuah JSON-encoded string yang mewakili daftar fungsi agregasi untuk diterapkan. (Jika Anda tidak ingin agregasi, tentukan`UNAGGREGATED`.)
- `useNewDataFrame`— Jika benar, hasil dari GROUP_BY tersedia dalam sesi proyek, menggantikan konten saat ini.

Example Contoh

```
[
  {
    "Action": {
      "Operation": "GROUP_BY",
      "Parameters": {
        "groupByAggFunctionOptions": "[{\"sourceColumnName\":\"all_votes\",
\"targetColumnName\":\"all_votes_count\",\"targetColumnDataType\":\"number\",
\"functionName\":\"COUNT\"}]",
        "sourceColumns": "[\"year\",\"state_name\"]",
        "useNewDataFrame": "true"
      }
    }
  }
]
```

BERGABUNG

Melakukan operasi gabungan pada dua kumpulan data.

Parameter

- `joinKeys`— JSON-encoded String yang mewakili daftar kolom dari setiap dataset untuk bertindak sebagai kunci gabungan.
- `joinType`— Jenis bergabung untuk tampil. Harus menjadi salah satu dari: `INNER_JOIN` `LEFT_JOIN` | `RIGHT_JOIN` | `OUTER_JOIN` | `LEFT_EXCLUDING_JOIN` | `RIGHT_EXCLUDING_JOIN` | `OUTER_EXCLUDING_JOIN`
- `leftColumns`— JSON-encoded String yang mewakili daftar kolom dari dataset aktif saat ini.

- `rightColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom dari dataset lain (sekunder) untuk bergabung dengan yang sekarang.
- `secondInputLocation`— URL Amazon S3 yang menyelesaikan file data untuk kumpulan data sekunder.
- `secondaryDatasetName`— Nama dataset sekunder.

Example Contoh

```
{
    "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"
        }
    }
}
```

POROS

Mengkonversi semua nilai baris dalam kolom yang dipilih menjadi kolom individual dengan nilai.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom dapat memiliki maksimum 10 nilai yang berbeda.
- `valueColumn`— Nama kolom yang ada. Kolom dapat memiliki maksimum 10 nilai yang berbeda.
- `aggregateFunction`— Nama fungsi agregasi. Jika Anda tidak ingin agregasi, gunakan kata kunci`COLLECT_LIST`.

Example Contoh

```
{
    "Action": {
        "Operation": "PIVOT",
        "Parameters": {
            "aggregateFunction": "SUM",
            "sourceColumn": "state_name",
            "valueColumn": "all_votes"
        }
    }
}
```

SKALA

Menimbang atau menormalkan rentang data dalam kolom numerik.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `strategy`— Operasi yang akan diterapkan pada nilai kolom:
  - `MIN_MAX`— Mengubah nilai menjadi kisaran [0,1].
  - `SCALE_BETWEEN`— Mengubah nilai ke dalam rentang dua nilai yang ditentukan.
  - `MEAN_NORMALIZATION`— Mengubah skala data untuk memiliki rata-rata (μ) 0 dan standar deviasi (σ) dari 1 dalam kisaran [-1, 1].
  - `Z_SCORE`— Nilai data skala linier memiliki rata-rata (μ) 0 dan standar deviasi (σ) 1. Terbaik untuk menangani outlier.
- `targetColumn`— Nama kolom yang berisi hasil.

Example Contoh

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

MENTRANSPOS

Mengkonversi semua baris yang dipilih ke kolom dan kolom ke baris.

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

Parameter

- `pivotColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang barisnya akan dikonversi ke nama kolom.
- `valueColumns`— Sebuah JSON-encoded string yang mewakili daftar satu atau lebih kolom untuk dikonversi ke baris.
- `aggregateFunction`— Nama fungsi agregasi. Jika Anda tidak ingin agregasi, gunakan kata kunci`COLLECT_LIST`.
- `newColumn`- Kolom untuk menahan kolom yang dialihkan sebagai nilai.

Example Contoh

```
{
    "Action": {
        "Operation": "TRANSPOSE",
        "Parameters": {
            "pivotColumns": "[\"Teacher\"]",
            "valueColumns": "[\"Tom\",\"John\",\"Harry\"]",
            "aggregateFunction": "COLLECT_LIST",
            "newColumn": "Student"
        }
    }

}
```

UNION

Menggabungkan baris dari dua atau lebih kumpulan data menjadi satu hasil.

Parameter

- `datasetsColumns`— Sebuah JSON-encoded string yang mewakili daftar semua kolom dalam dataset.
- `secondaryDatasetNames`— JSON-encoded String yang mewakili daftar satu atau lebih kumpulan data sekunder.
- `secondaryInputs`— JSON-encoded String yang mewakili daftar bucket Amazon S3 dan nama kunci objek yang memberi tahu DataBrew di mana menemukan kumpulan data sekunder.
- `targetColumnNames`— Sebuah JSON-encoded string yang mewakili daftar nama kolom untuk hasil.

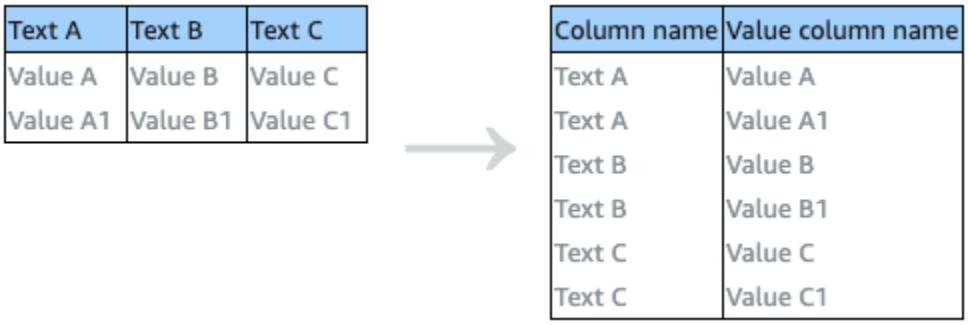
Example Contoh

```
{
    "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

Mengkonversi semua nilai kolom dalam baris yang dipilih menjadi baris individual dengan nilai.



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

Parameter

- `sourceColumns`— JSON-encoded String yang mewakili daftar satu atau lebih kolom yang tidak diputar.
- `unpivotColumn`- Kolom nilai untuk operasi unpivot.
- `valueColumn`- Kolom untuk menyimpan nilai unpivoted.

Example Contoh

```
{
    "Action": {
        "Operation": "UNPIVOT",
```

```
        "Parameters": {
            "sourceColumns": "[\"idealpoint_estimate\"]",
            "unpivotColumn": "unpivoted_idealpoint_estimate",
            "valueColumn": "unpivoted_column_values"
        }
    }
}
```

Langkah-langkah resep ilmu data

Gunakan langkah-langkah resep ini untuk mentabulasi dan meringkas data dari perspektif yang berbeda, atau untuk melakukan transformasi lanjutan.

Topik

- BINARISASI
- BUCKETIZATION
- CATEGORICAL_MAPPING
- ONE_HOT_PENGKODEAN
- SKALA
- KEMIRINGAN
- TOKENISASI

BINARISASI

Mengambil semua nilai dalam kolom sumber numerik yang dipilih, membandingkannya dengan nilai ambang batas, dan mengeluarkan kolom baru dengan 1 atau 0 untuk setiap baris.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `targetColumn`— Nama kolom baru yang akan dibuat.

  `threshold`— Angka yang menunjukkan ambang batas untuk menetapkan nilai 0 atau 1.

  `flip`— Opsi untuk membalik tugas biner sehingga nilai yang lebih rendah diberikan 1 dan nilai yang lebih tinggi diberikan 0. Ketika parameter flip benar, nilai yang lebih rendah dari atau sama dengan nilai ambang menghasilkan 1, dan nilai yang lebih besar dari nilai ambang menghasilkan 0.

Example Contoh

```
{
    "Action": {
        "Operation": "BINARIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "threshold": "100.0",
            "flip": "false"
        }
    }
}
```

BUCKETIZATION

Bucketization (disebut Binning di konsol) mengambil item dalam kolom nilai numerik, mengelompokkannya ke dalam bin yang ditentukan oleh rentang numerik, dan mengeluarkan kolom baru yang menampilkan bin untuk setiap baris. Bucketization dapat dilakukan dengan menggunakan split atau persentase. Contoh pertama di bawah ini menggunakan split dan contoh kedua menggunakan persentase.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `targetColumn`— Nama kolom baru yang akan dibuat.

  `bucketNames`— Daftar nama ember.

  `splits`— Daftar level bucket. Ember berurutan, dan batas atas untuk ember akan menjadi batas bawah untuk ember berikutnya.

  `percentage`— Setiap ember akan digambarkan sebagai persentase.

Example Contoh menggunakan split

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

```
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\",\"Bin3\"]",
            "splits": "[\"-Infinity\",\"2\",\"20\",\"Infinity\"]"
        }
    }
}
```

Example Contoh menggunakan persentase

```
{
    "Action": {
        "Operation": "BUCKETIZATION",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "bucketNames": "[\"Bin1\",\"Bin2\"]",
            "percentage": "50"
        }
    }
}
```

CATEGORICAL_MAPPING

Memetakan satu atau lebih nilai kategoris ke nilai numerik atau lainnya

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `categoryMap`— JSON-encoded String yang mewakili peta nilai ke kategori.

  `deleteOtherRows`— Jika`true`, semua baris yang tidak dipetakan akan dihapus dari kumpulan data.

  `other`— Saat disediakan, semua nilai yang tidak dipetakan akan diganti dengan nilai ini.

  `keepOthers`— Jika benar, semua nilai yang tidak dipetakan akan tetap sama.

  `mapType`— Tipe data dari kolom yang dipetakan.

`targetColumn`— Nama kolom yang berisi hasil.

Example Contoh

```
{
    "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_PENGKODEAN

Menciptakan n kolom numerik, di mana n adalah jumlah nilai unik dalam variabel kategoris yang dipilih.

Misalnya, pertimbangkan kolom bernama`shirt_size`. Kemeja tersedia dalam ukuran kecil, sedang, besar, atau ekstra besar. Data kolom mungkin terlihat seperti berikut ini.

```
shirt_size
-----------
L
XL
M
S
M
M
S
XL
M
L
XL
```

```
M
```

Dalam skenario ini, ada empat nilai berbeda untuk`shirt_size`. Oleh karena itu, `ONE_HOT_ENCODING` menghasilkan empat kolom baru. Setiap kolom baru diberi nama`shirt_size_`*x*, di mana *x* mewakili `shirt_size` nilai yang berbeda.

Hasil `shirt_size` dan empat kolom yang dihasilkan terlihat seperti ini.

```
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
```

Kolom yang Anda tentukan `ONE_HOT_ENCODING` dapat memiliki maksimum sepuluh (10) nilai berbeda.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom dapat memiliki maksimum 10 nilai yang berbeda.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ONE_HOT_ENCODING",
        "Parameters": {
            "sourceColumn": "shirt_size"
        }
    }
}
```

SKALA

Menimbang atau menormalkan rentang data dalam kolom numerik.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `strategy`— Operasi yang akan diterapkan pada nilai kolom:
  - `MIN_MAX`— Mengubah nilai menjadi kisaran [0,1]
  - `SCALE_BETWEEN`— Rescales nilai ke dalam kisaran 2 nilai yang ditentukan.
  - `MEAN_NORMALIZATION`— Mengubah skala data untuk memiliki rata-rata (μ) 0 dan standar deviasi (σ) dari 1 dalam kisaran [-1, 1]
  - `Z_SCORE`— Nilai data skala linier memiliki rata-rata (μ) 0 dan standar deviasi (σ) 1. Terbaik untuk menangani outlier.
- `targetColumn`— Nama kolom yang berisi hasil.

Example Contoh

```
{
    "Action": {
        "Operation": "NORMALIZATION",
        "Parameters": {
            "sourceColumn": "all_votes",
            "strategy": "MIN_MAX",
            "targetColumn": "all_votes_normalized"
        }
    }
}
```

KEMIRINGAN

Menerapkan transformasi pada nilai data Anda untuk mengubah bentuk distribusi dan kemiringannya.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `targetColumn`— Nama kolom baru yang akan dibuat.

`skewFunction`

- `ROOT`— ekstrak nilai-root. Root dapat disediakan dalam `value` parameter.

  `LOG`— nilai dasar log. Basis log dapat disediakan dalam `value` parameter.

  `SQUARE`— fungsi persegi

`value`— Argumen dari SkewFunction.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SKEWNESS",
        "Parameters": {
            "sourceColumn": "level",
            "targetColumn": "bin",
            "skewFunction": "LOG",
            "value": "2.718281828"
        }
    }
}
```

TOKENISASI

Membagi teks menjadi unit yang lebih kecil, atau token, seperti kata atau istilah individual.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `delimiter`— Sebuah pembatas kustom yang muncul di antara kata-kata tokenized. (Perilaku default adalah memisahkan setiap token dengan spasi.)
- `expandContractions`— Jika`ENABLED`, memperluas kata-kata yang dikontrak. Misalnya: “jangan” menjadi “jangan”.
- `stemmingMode`Membagi teks menjadi unit atau token yang lebih kecil, seperti kata atau istilah huruf kecil individual. Dua mode stemming tersedia: `PORTER` |. `LANCASTER`
- `stopWordRemovalMode`— Menghapus kata-kata umum seperti a, an, the, dan banyak lagi.

- `customStopWords`— Untuk`StopWordRemovalMode`, memungkinkan Anda untuk menentukan daftar kustom kata-kata berhenti.
- `targetColumn`— Nama kolom yang berisi hasil.

Example Contoh

```
{
    "Action": {
        "Operation": "TOKENIZATION",
        "Parameters": {
            "customStopWords": "[]",
            "delimiter": "- ",
            "expandContractions": "ENABLED",
            "sourceColumn": "dimensions",
            "stemmingMode": "PORTER",
            "stopWordRemovalMode": "DEFAULT",
            "targetColumn": "dimensions_tokenized"
        }
    }

}
```

Fungsi matematika

Berikut, temukan topik referensi untuk fungsi matematika yang bekerja dengan tindakan resep.

Topik

- MUTLAK
- MENAMBAHKAN
- LANGIT-LANGIT
- GELAR
- MEMBAGI
- EKSPONEN
- FLOOR
- ADALAH_GENAP
- ADALAH_ANEH

- LN
- LOG
- MOD
- KALIKAN
- MENIADAKAN
- PI
- DAYA
- RADIAN
- ACAK
- RANDOM_BETWEEN
- BULAT
- TANDA
- SQUARE_ROOT
- KURANGI

MUTLAK

Mengembalikan nilai absolut dari nomor masukan dalam kolom baru. Nilai absolut adalah seberapa jauh angkanya dari nol, terlepas dari apakah itu positif atau negatif

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ABSOLUTE",
        "Parameters": {
            "sourceColumn": "freezingTemps",
            "targetColumn": "absValueOfFreezingTemps"
```

```
        }
    }
}
```

MENAMBAHKAN

Jumlahkan nilai kolom input di kolom baru, menggunakan (`sourceColumn1+sourceColumn2`) atau (`sourceColumn1+value1`).

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `value1`— Nilai numerik.
- `sourceColumn2`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ADD",
        "Parameters": {
            "sourceColumn1": "weight_kg",
            "sourceColumn2": "height_cm",
            "targetColumn": "weight_plus_height"
        }
    }
}
```

LANGIT-LANGIT

Mengembalikan bilangan bulat terkecil yang lebih besar dari atau sama dengan angka desimal masukan di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value1`— Nilai numerik.

- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "CEILING",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_CEILING"
        }
    }
}
```

GELAR

Mengkonversi radian untuk sudut ke derajat dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DEGREES",
        "Parameters": {
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_DEGREES"
        }
    }
}
```

MEMBAGI

Membagi satu nomor input dengan yang lain dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `value1`— Nilai numerik.
- `sourceColumn2`— Nama kolom yang ada.
- `value2`— Nilai numerik.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DIVIDE",
        "Parameters": {
            "sourceColumn1": "height_cm",
            "targetColumn": "divide_by_2",
            "value2": "2"
        }
    }
}
```

EKSPONEN

Mengembalikan nomor Euler dinaikkan ke derajat n th dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "EXPONENT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_EXPONENT"
```

```
        }
    }
}
```

FLOOR

Mengembalikan bilangan integral terbesar yang lebih besar dari atau sama dengan nomor masukan di kolom baru.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `value`— Nilai numerik.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FLOOR",
        "Parameters": {
            "targetColumn": "FLOOR Column 1",
            "value": "42"
        }
    }
}
```

ADALAH_GENAP

Mengembalikan nilai Boolean di kolom baru yang menunjukkan apakah kolom sumber atau nilai genap. Jika kolom sumber atau nilai adalah desimal, hasilnya salah.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `trueString`— String yang menunjukkan apakah nilainya genap.
- `falseString`— String yang menunjukkan apakah nilainya tidak genap.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "IS_EVEN",
        "Parameters": {
            "falseString": "Value is odd",
            "sourceColumn": "height_cm",
            "targetColumn": "height_cm_IS_EVEN",
            "trueString": "Value is even"
        }
    }
}
```

ADALAH_ANEH

Mengembalikan nilai Boolean di kolom baru yang menunjukkan apakah kolom sumber atau nilai ganjil. Jika kolom sumber atau nilai adalah desimal, hasilnya salah.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `trueString`— String yang menunjukkan apakah nilainya ganjil.
- `falseString`— String yang menunjukkan apakah nilainya tidak ganjil.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "IS_ODD",
        "Parameters": {
            "falseString": "Value is even",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_IS_ODD",
            "trueString": "Value is odd"
        }
    }
}
```

LN

Mengembalikan logaritma natural (nomor Euler) dari nilai dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "LN",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_LN"
        }
    }
}
```

LOG

Mengembalikan logaritma nilai dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `base`— Dasar logaritma. Default-nya adalah 10.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "LOG",
        "Parameters": {
            "base": "10",
```

```
            "sourceColumn": "age",
            "targetColumn": "age_LOG"
        }
    }
}
```

MOD

Mengembalikan persen bahwa satu nomor adalah nomor lain di kolom baru.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MOD",
        "Parameters": {
            "sourceColumn1": "start_date",
            "sourceColumn2": "end_date",
            "targetColumn": "MOD Column 1"
        }
    }
}
```

KALIKAN

Mengalikan dua angka dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `value1`— Nilai numerik.
- `sourceColumn2`— Nama kolom yang ada.
- `value2`— Nilai numerik.

- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MULTIPLY",
        "Parameters": {
            "sourceColumn1": "hourly_rate",
            "sourceColumn2": "hours",
            "targetColumn": "total_pay"
        }
    }
}
```

MENIADAKAN

Menegasikan nilai dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NEGATE",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_NEGATE"
        }
    }
}
```

PI

Mengembalikan nilai pi (3.141592653589793) di kolom baru.

Parameter

- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "PI",
        "Parameters": {
            "targetColumn": "PI Column 1"
        }
    }
}
```

DAYA

Mengembalikan nilai angka ke kekuatan eksponen di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— Angka yang nilainya akan dinaikkan.
- `targetColumn`— Nama kolom baru yang akan dibuat.
- `exponent`— Kekuatan yang nilainya akan dinaikkan.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

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

```
            "exponent": "3",
            "sourceColumn": "age",
            "targetColumn": "age_cubed"
        }
    }
}
```

RADIAN

Mengkonversi derajat ke radian (dibagi dengan 180/pi) dan mengembalikan nilai dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "RADIANS",
        "Parameters": {
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_RADIANS"
        }
    }
}
```

ACAK

Mengembalikan angka acak antara 0 dan 1 di kolom baru.

Parameter

- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
```

```
    "RecipeAction": {
        "Operation": "RANDOM",
        "Parameters": {
            "targetColumn": "RANDOM Column 1"
        }
    }
}
```

RANDOM_BETWEEN

Di kolom baru, mengembalikan nomor acak antara batas bawah tertentu (inklusif) dan batas atas tertentu (inklusif).

Parameter

- `lowerBound`— Batas bawah dari rentang angka acak.
- `upperBound`— Batas atas rentang angka acak.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "RANDOM_BETWEEN",
        "Parameters": {
            "lowerBound": "1",
            "targetColumn": "RANDOM_BETWEEN Column 1",
            "upperBound": "100"
        }
    }
}
```

BULAT

Membulatkan nilai numerik ke bilangan bulat terdekat di kolom baru. Ini membulatkan ketika fraksi 0,5 atau lebih.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ROUND",
        "Parameters": {
            "sourceColumn": "rating",
            "targetColumn": "rating_ROUND"
        }
    }
}
```

TANDA

Mengembalikan kolom baru dengan -1 jika nilainya kurang dari 0, 0 jika nilainya 0, dan +1 jika nilainya lebih besar dari 0.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SIGN",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SIGN"
        }
    }
}
```

SQUARE_ROOT

Mengembalikan akar kuadrat dari nilai dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SQUARE_ROOT",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_SQUARE_ROOT"
        }
    }
}
```

KURANGI

Mengurangi satu angka dari yang lain dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `value1`— Nilai numerik.
- `sourceColumn2`— Nama kolom yang ada.
- `value2`— Nilai numerik.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SUBTRACT",
        "Parameters": {
            "sourceColumn1": "weight_kg",
```

```
            "targetColumn": "weight_minus_10_kg",
            "value2": "10"
        }
    }
}
```

Fungsi agregat

Mengikuti, temukan topik referensi untuk fungsi agregat yang bekerja dengan tindakan resep.

Topik

- APA PUN
- RATA-RATA
- COUNT
- COUNT_DISTINCT
- KTH_LARGEST
- KTH_LARGEST_UNIQUE
- MAX
- MEDIAN
- MIN
- MODE
- STANDARD_DEVIASI
- JUMLAH
- PERBEDAAN

APA PUN

Mengembalikan nilai apapun dari kolom sumber yang dipilih dalam kolom baru. Nilai kosong dan nol diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ANY",
        "Parameters": {
            "sourceColumns": "[\"age\",\"last_name\"]",
            "targetColumn": "ANY Column 1"
        }
    }
}
```

RATA-RATA

Menghitung rata-rata nilai di kolom sumber dan mengembalikan hasilnya di kolom baru. Non-nomor apa pun diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "AVERAGE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"weight_kg\",\"height_cm\"]",
            "targetColumn": "AVERAGE Column 1"
        }
    }
}
```

COUNT

Mengembalikan jumlah nilai dari kolom sumber yang dipilih di kolom baru. Nilai kosong dan nol diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "COUNT",
        "Parameters": {
            "sourceColumns": "[\"ANY Column 1\",\"birth_date\",\"last_name\"]",
            "targetColumn": "COUNT Column 1"
        }
    }
}
```

COUNT_DISTINCT

Mengembalikan jumlah total nilai yang berbeda dari kolom sumber yang dipilih di kolom baru. Nilai kosong dan nol diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "COUNT_DISTINCT",
        "Parameters": {
            "sourceColumns": "[\"long_name\",\"weight_kg\"]",
            "targetColumn": "COUNT_DISTINCT Column 1"
        }
    }
}
```

KTH_LARGEST

Mengembalikan nomor terbesar k th dari kolom sumber yang dipilih di kolom baru.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.
- `value`— Angka yang mewakili k.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "KTH_LARGEST",
        "Parameters": {
            "sourceColumns": "[\"height_cm\",\"weight_kg\",\"age\"]",
            "targetColumn": "KTH_LARGEST Column 1",
            "value": "2"
        }
    }
}
```

KTH_LARGEST_UNIQUE

Mengembalikan nomor unik k th terbesar dari kolom sumber yang dipilih di kolom baru.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

  `value`— Angka yang mewakili k.

Example Contoh

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

```
        "Operation": "KTH_LARGEST_UNIQUE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "KTH_LARGEST_UNIQUE Column 1",
            "value": "3"
        }
    }
}
```

MAX

Mengembalikan nilai numerik maksimum dari kolom sumber yang dipilih di kolom baru. Non-nomor apa pun diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MAX",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MAX Column 1"
        }
    }
}
```

MEDIAN

Mengembalikan median, nomor tengah dari kelompok angka yang diurutkan, dari kolom sumber yang dipilih di kolom baru. Non-nomor apa pun diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MEDIAN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "MEDIAN Column 1"
        }
    }
}
```

MIN

Mengembalikan nilai minimum dari kolom sumber yang dipilih di kolom baru. Non-nomor apa pun diabaikan.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MIN",
        "Parameters": {
            "sourceColumns": "[\"age\",\"height_cm\",\"weight_kg\"]",
            "targetColumn": "MIN Column 1"
        }
    }
}
```

MODE

Mengembalikan mode, nomor yang paling sering muncul, dari kolom sumber yang dipilih di kolom baru. Non-nomor apa pun diabaikan. Untuk beberapa mode, mode dihitung dengan fungsi modal.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "sourceColumns": "[\"years_in_service\",\"age\"]",
            "targetColumn": "MODE Column 1"
        }
    }
}
```

STANDARD_DEVIASI

Mengembalikan standar deviasi dari kolom sumber yang dipilih di kolom baru.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "STANDARD_DEVIATION",
        "Parameters": {
            "sourceColumns": "[\"years_in_sservice\",\"age\"]",
            "targetColumn": "STANDARD_DEVIATION Column 1"
        }
    }
}
```

JUMLAH

Mengembalikan jumlah nilai dari kolom sumber yang dipilih di kolom baru. Setiap non-angka diperlakukan sebagai 0.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SUM",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
            "targetColumn": "SUM Column 1"
        }
    }
}
```

PERBEDAAN

Mengembalikan varians dari kolom sumber yang dipilih di kolom baru. Varians didefinisikan sebagai`Var(X) = [Sum ((X – mean(X))^2)]/Count(X)`.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "VARIANCE",
        "Parameters": {
            "sourceColumns": "[\"age\",\"years_in_service\"]",
```

```
            "targetColumn": "VARIANCE Column 1"
        }
    }
}
```

Fungsi teks

Mengikuti, temukan topik referensi untuk fungsi teks yang bekerja dengan tindakan resep.

Topik

- CHAR
- ENDS_WITH
- PASTI
- TEMUKAN
- KIRI
- LEN
- LEBIH RENDAH
- MERGE_COLUMNS_AND_VALUES
- TEPAT
- REMOVE_SYMBOLS
- REMOVE_WHITESPACE
- REPEAT_STRING
- KANAN
- RIGHT_FIND
- STARTS_WITH
- STRING_GREATER_THAN
- STRING_GREATER_THAN_EQUAL
- STRING_LESS_THAN
- STRING_LESS_THAN_EQUAL
- SUBSTRING
- MEMANGKAS
- UNICODE
- ATAS

CHAR

Mengembalikan dalam kolom baru karakter Unicode untuk setiap integer di kolom sumber, atau untuk nilai integer kustom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— Integer yang mewakili nilai Unicode.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "sourceColumn": "age",
            "targetColumn": "age_char"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "CHAR",
        "Parameters": {
            "value": 42,
            "targetColumn": "asterisk"
        }
    }
}
```

ENDS_WITH

Kembali `true` dalam kolom baru jika jumlah tertentu karakter paling kanan, atau string kustom, cocok dengan pola.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `pattern`— Ekspresi reguler yang harus cocok dengan akhir string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "ENDS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[Ss]",
            "targetColumn": "nationality_ends_with"
        }
    }
}
```

PASTI

Membuat kolom baru diisi dengan salah satu dari berikut ini:

- `True`jika satu string dalam kolom (atau nilai) sama persis dengan string lain di kolom (atau nilai) yang berbeda.
- `False`jika tidak ada kecocokan.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.
- `value2`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda hanya dapat menentukan satu dari kombinasi berikut:
>
> - Keduanya`sourceColumn`*N*.
> - Salah satu `sourceColumn`*N* dan satu dari`value`*N*.
> - Keduanya`value`*N*.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "EXACT",
        "Parameters": {
            "sourceColumn1": "nationality",
            "value2": "Argentina",
            "targetColumn": "nationality_exact"
        }
    }
}
```

TEMUKAN

Mencari dari kiri ke kanan, menemukan string yang cocok dengan string tertentu dari kolom sumber atau dari nilai kustom, dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `pattern`— Ekspresi reguler untuk dicari.
- `position`— Posisi karakter untuk memulai, dari ujung kiri string.
- `ignoreCase`— Jika`true`, abaikan perbedaan huruf (antara huruf besar dan kecil) di antara huruf. Untuk menegakkan pencocokan yang ketat, gunakan `false` sebagai gantinya.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "FIND",
        "Parameters": {
            "sourceColumn": "city",
            "pattern": "[AEIOU]",
            "position": "1",
            "ignoreCase": "false",
            "targetColumn": "begins_with_a_vowel"
        }
    }
}
```

KIRI

Diberikan sejumlah karakter, mengambil jumlah karakter paling kiri dalam string dari kolom sumber atau string kustom, dan mengembalikan jumlah karakter paling kiri yang ditentukan di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `position`— Posisi karakter untuk memulai, dari ujung kiri string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Mengembalikan dalam kolom baru panjang string dari kolom sumber atau string kustom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_len"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LEN",
        "Parameters": {
            "value": "Hello",
            "targetColumn": "hello_len"
        }
    }
}
```

LEBIH RENDAH

Mengkonversi semua karakter abjad dari string di kolom sumber atau string kustom ke huruf kecil, dan mengembalikan hasilnya dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_lower"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "LOWER",
        "Parameters": {
            "value": "GOODBYE",
            "targetColumn": "goodbye_lower"
        }
    }
}
```

MERGE_COLUMNS_AND_VALUES

Menggabungkan string di kolom sumber dan mengembalikan hasilnya di kolom baru. Anda dapat menyisipkan pembatas antara nilai gabungan.

Parameter

- `sourceColumns`— Nama dua atau lebih kolom yang ada, dalam JSON-encoded format.
- `delimiter` – Opsional. Satu atau lebih karakter untuk ditempatkan di antara masing-masing dua nilai kolom sumber.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
```

```
    "RecipeAction": {
        "Operation": "MERGE_COLUMNS_AND_VALUES",
        "Parameters": {
            "sourceColumns": "[\"last_name\",\"birth_date\"]",
            "delimiter": " was born on: ",
            "targetColumn": "merged_column"
        }
    }
}
```

TEPAT

Mengkonversi semua karakter abjad dari string di kolom sumber atau nilai kustom ke kasus yang tepat, dan mengembalikan hasilnya dalam kolom baru.

Dalam kasus yang tepat, juga disebut huruf kapital, huruf pertama dari setiap kata dikapitalisasi dan sisa kata diubah menjadi huruf kecil. Contohnya adalah: Rubah Coklat Cepat Melompati Pagar

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Menghapus karakter yang bukan huruf, angka, karakter Latin beraksen, atau spasi putih dari string di kolom sumber atau string kustom, dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Menghapus spasi putih dari string di kolom sumber atau string kustom, dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Mengulangi string di kolom sumber atau nilai input kustom beberapa kali tertentu, dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `count`— Berapa kali untuk mengulangi string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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"
        }
    }
}
```

KANAN

Diberikan sejumlah karakter, mengambil jumlah karakter paling kanan dalam string dari kolom sumber atau string kustom, dan mengembalikan jumlah karakter paling kanan yang ditentukan di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `position`— Posisi karakter untuk memulai, dari sisi kanan string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Mencari dari kanan ke kiri, menemukan string yang cocok dengan string tertentu dari kolom sumber atau dari nilai kustom, dan mengembalikan hasilnya di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `pattern`— Ekspresi reguler untuk dicari.
- `position`— Posisi karakter untuk memulai, dari ujung kanan string.
- `ignoreCase`— Jika`true`, abaikan perbedaan huruf (antara huruf besar dan kecil) di antara huruf. Untuk menegakkan pencocokan yang ketat, gunakan `false` sebagai gantinya.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

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

```
        "Operation": "RIGHT_FIND",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "s",
            "position": "1",
            "ignoreCase": "true",
            "targetColumn": "ends_with_an_s"
        }
    }
}
```

STARTS_WITH

Kembali `true` dalam kolom baru jika jumlah tertentu karakter paling kiri, atau string kustom, cocok dengan pola.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `pattern`— Ekspresi reguler yang harus cocok dengan awal string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "STARTS_WITH",
        "Parameters": {
            "sourceColumn": "nationality",
            "pattern": "[AEIOU]",
            "targetColumn": "nationality_starts_with"
        }
    }
```

```
}
```

STRING_GREATER_THAN

Membuat kolom baru diisi dengan salah satu dari berikut ini:

- `True`jika satu string dalam kolom (atau nilai) lebih besar dari string lain di kolom yang berbeda (atau nilai).
- `False`jika tidak ada kecocokan.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.
- `value2`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda hanya dapat menentukan satu dari kombinasi berikut:
>
> - Keduanya`sourceColumn`*N*.
> - Salah satu `sourceColumn`*N* dan satu dari`value`*N*.
> - Keduanya`value`*N*.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN",
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_greater_than"
```

```
        }
    }
}
```

STRING_GREATER_THAN_EQUAL

Membuat kolom baru diisi dengan salah satu dari berikut ini:

- `True`jika satu string dalam kolom (atau nilai) lebih besar dari atau sama dengan string lain di kolom yang berbeda (atau nilai).
- `False`jika tidak ada kecocokan.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.
- `value2`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda hanya dapat menentukan satu dari kombinasi berikut:
>
> - Keduanya`sourceColumn`*N*.
> - Salah satu `sourceColumn`*N* dan satu dari`value`*N*.
> - Keduanya`value`*N*.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "STRING_GREATER_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "nationality",
```

```
            "targetColumn": "string_greater_than_equal",
            "value2": "s"
        }
    }
}
```

STRING_LESS_THAN

Membuat kolom baru diisi dengan salah satu dari berikut ini:

- `True`jika satu string dalam kolom (atau nilai) kurang dari string lain di kolom yang berbeda (atau nilai).
- `False`jika tidak ada kecocokan.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.
- `value2`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda hanya dapat menentukan satu dari kombinasi berikut:
>
> - Keduanya`sourceColumn`*N*.
> - Salah satu `sourceColumn`*N* dan satu dari`value`*N*.
> - Keduanya`value`*N*.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN",
```

```
        "Parameters": {
            "sourceColumn1": "first_name",
            "sourceColumn2": "last_name",
            "targetColumn": "string_less_than"
        }
    }
}
```

STRING_LESS_THAN_EQUAL

Membuat kolom baru diisi dengan salah satu dari berikut ini:

- `True`jika satu string dalam kolom (atau nilai) kurang dari atau sama dengan string lain di kolom yang berbeda (atau nilai).
- `False`jika tidak ada kecocokan.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.
- `value2`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda hanya dapat menentukan satu dari kombinasi berikut:
>
> - Keduanya`sourceColumn`*N*.
> - Salah satu `sourceColumn`*N* dan satu dari`value`*N*.
> - Keduanya`value`*N*.

Example Contoh

```
{
```

```
    "RecipeAction": {
        "Operation": "STRING_LESS_THAN_EQUAL",
        "Parameters": {
            "sourceColumn1": "first_name",
            "targetColumn": "string_less_than_equal",
            "value2": "s"
        }
    }
}
```

SUBSTRING

Mengembalikan dalam kolom baru beberapa atau semua string yang ditentukan di kolom sumber, berdasarkan nilai indeks awal dan akhir yang ditentukan pengguna.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `startPosition`— Posisi karakter untuk memulai, dari ujung kiri string.
- `endPosition`— Posisi karakter untuk diakhiri dengan, dari ujung kiri string.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SUBSTRING",
        "Parameters": {
            "sourceColumn": "last_name",
            "startPosition": "5",
            "endPosition": "8",
            "targetColumn": "chars_5_through_8"
        }
    }
```

```
}
```

MEMANGKAS

Menghapus spasi putih depan dan belakang dari string di kolom sumber atau string kustom, dan mengembalikan hasilnya di kolom baru. Spasi di antara kata-kata tidak dihapus.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "sourceColumn": "nationality",
            "targetColumn": "nationality_trim"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "TRIM",
        "Parameters": {
            "value": "   This string should be trimmed       ",
            "targetColumn": "string_trimmed"
        }
```

```
    }
}
```

UNICODE

Mengembalikan dalam kolom baru nilai indeks Unicode untuk karakter pertama string di kolom sumber atau untuk string kustom.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "sourceColumn": "first_name",
            "targetColumn": "first_name_unicode"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UNICODE",
        "Parameters": {
            "value": "?",
            "targetColumn": "sixty_three"
```

```
        }
    }
}
```

ATAS

Mengkonversi semua karakter abjad dari string di kolom sumber atau string kustom ke huruf besar, dan mengembalikan hasilnya dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "sourceColumn": "last_name",
            "targetColumn": "last_name_upper"
        }
    }
}
```

```
{
    "RecipeAction": {
        "Operation": "UPPER",
        "Parameters": {
            "value": "a string of lowercase letters",
            "targetColumn": "string_upper"
```

```
        }
    }
}
```

Fungsi tanggal dan waktu

Berikut, temukan topik referensi untuk fungsi tanggal dan waktu yang bekerja dengan tindakan resep.

Topik

- CONVERT_TIMEZONE
- DATE
- DATE_ADD
- DATE_DIFF
- DATE_FORMAT
- DATE_TIME
- DAY
- HOUR
- MILIDETIK
- MINUTE
- MONTH
- BULAN_NAMA
- SEKARANG
- KUARTAL
- DETIK
- TIME
- HARI INI
- UNIX_WAKTU
- FORMAT UNIX_TIME_
- WEEK_DAY
- WEEK_NUMBER
- YEAR

CONVERT_TIMEZONE

Mengkonversi nilai waktu dari kolom sumber ke kolom baru berdasarkan zona waktu tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom sumber dapat berupa tipe`string`,`date`, atau`timestamp`.
- `fromTimeZone`— Zona waktu nilai sumber. Jika tidak ada yang ditentukan, zona waktu default adalah UTC.
- `toTimeZone`— Zona waktu untuk dikonversi ke. Jika tidak ada yang ditentukan, zona waktu default adalah UTC.
- `targetColumn`— Nama untuk kolom yang baru dibuat.
- `dateTimeFormat` – Opsional. String format untuk tanggal. Jika format tidak ditentukan, format default digunakan:`yyyy-mm-dd HH:MM:SS`.

Example Contoh

```
{
    "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

Membuat kolom baru yang berisi nilai tanggal, dari kolom sumber atau dari nilai yang disediakan.

Parameter

- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika string ini tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.

- `dateTimeParameters`— JSON-encoded String yang mewakili komponen tanggal dan waktu:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Setiap komponen harus menentukan salah satu dari berikut ini:
  - `sourceColumn`— Nama kolom yang ada.
  - `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "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

Menambahkan tahun, bulan, atau hari ke tanggal dari kolom sumber atau nilai, dan membuat kolom baru yang berisi hasil.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.

- `units`— Satuan ukuran untuk menyesuaikan tanggal. Nilai yang valid adalah `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, dan`MINUTES`.
- `dateAddValue`— Jumlah yang `units` akan ditambahkan ke tanggal.
- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DATE_ADD",
        "Parameters": {
            "sourceColumn": "DATE Column 1",
            "units": "DAYS",
            "dateAddValue": "14",
            "dateTimeFormat": "mm/dd/yyyy",
            "targetColumn": "DATE Column 1_DATEADD"
        }
    }
}
```

DATE_DIFF

Membuat kolom baru yang berisi perbedaan antara dua tanggal.

Parameter

- `sourceColumn1`— Nama kolom yang ada.
- `sourceColumn2`— Nama kolom yang ada.
- `value1`— String karakter untuk dievaluasi.

- `value2`— String karakter untuk dievaluasi.
- `units`— Satuan ukuran untuk menggambarkan perbedaan antara tanggal. Nilai yang valid adalah `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, dan`MINUTES`.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda hanya dapat menentukan salah satu kombinasi berikut:
>
> - Keduanya `sourceColumn1` dan`sourceColumn2`.
> - Salah satu `sourceColumn1` atau `sourceColumn2` dan salah satu dari `value1` atau`value2`.
> - Keduanya `value1` dan`value2`.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DATE_DIFF",
        "Parameters": {
            "value1": "2020-01-01",
            "value2": "2020-10-06",
            "units": "DAYS",
            "targetColumn": "DATEDIFF Column 1"
        }
    }
}
```

DATE_FORMAT

Membuat kolom baru yang berisi tanggal, dalam format tertentu, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `value`— String untuk dievaluasi.
- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "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

Membuat kolom baru yang berisi nilai tanggal dan waktu, dari kolom sumber atau dari nilai yang disediakan.

Parameter

- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika string ini tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— JSON-encoded String yang mewakili komponen tanggal dan waktu:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Setiap komponen harus menentukan salah satu dari berikut ini:
  - `sourceColumn`— Nama kolom yang ada.
  - `value`— String karakter untuk dievaluasi.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DATE_TIME",
        "Parameters": {
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "dateTimeParameters": "{\"year\":{\"value\":\"2010\"},\"month\":{\"value\":
\"5\"},\"day\":{\"value\":\"21\"},\"hour\":{\"value\":\"13\"},\"minute\":{\"value\":
\"34\"},\"second\":{\"value\":\"25\"}}",
            "targetColumn": "DATETIME Column 1"
        }
    }
}
```

DAY

Membuat kolom baru yang berisi hari dalam sebulan, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_DAY"
        }
    }
}
```

HOUR

Membuat kolom baru yang berisi nilai jam, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "HOUR",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_HOUR"
        }
    }
}
```

MILIDETIK

Membuat kolom baru yang berisi nilai milidetik dari kolom sumber atau nilai input.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom sumber dapat berupa tipe`string`,`date`, atau`timestamp`.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MILLISECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MILLISECOND"
        }
    }
```

```
}
```

MINUTE

Membuat kolom baru yang berisi nilai menit, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MINUTE",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_MINUTE"
        }
    }
}
```

MONTH

Membuat kolom baru yang berisi jumlah bulan, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.

- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "MONTH",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTH Column 1"
        }
    }
}
```

BULAN_NAMA

Membuat kolom baru yang berisi nama bulan, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
```

```
    "RecipeAction": {
        "Operation": "MONTH_NAME",
        "Parameters": {
            "value": "2018-05-27",
            "targetColumn": "MONTHNAME Column 1"
        }
    }
}
```

SEKARANG

Membuat kolom baru yang berisi tanggal dan waktu saat ini dalam formatyyyy-mm-dd HH:MM:SS.

Parameter

- `timeZone`— Nama zona waktu. Jika tidak ada zona waktu yang ditentukan, maka defaultnya adalah Universal Coordinated Time (UTC).
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "NOW",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "NOW Column 1"
        }
}
```

KUARTAL

Membuat kolom baru yang berisi kuartal berbasis tanggal dari string yang mewakili tanggal.

> **Note**
>
> Kuartal ditetapkan di kolom baru sebagai 1, 2, 3, atau 4.
>
> - 1 Januari, Februari, dan Maret.
> - 2 adalah April, Mei, dan Juni.

- 3 adalah Juli, Agustus, dan September.
- 4 adalah Oktober, November, dan Desember.

Parameter

- `sourceColumn`— Nama kolom yang ada. Kolom sumber dapat berupa tipe`string`,`date`, atau`timestamp`.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "QUARTER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_QUARTER"
        }
    }
}
```

DETIK

Membuat kolom baru yang berisi nilai kedua, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "SECOND",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_SECOND"
        }
    }
}
```

TIME

Membuat kolom baru yang berisi nilai waktu, dari kolom sumber atau nilai yang disediakan.

Parameter

- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika string ini tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.
- `dateTimeParameters`— JSON-encoded String yang mewakili komponen tanggal dan waktu:
  - `year`
  - `value`
  - `month`
  - `day`
  - `hour`
  - `second`

  Setiap komponen harus menentukan salah satu dari berikut ini:
  - `sourceColumn`— Nama kolom yang ada.
  - `value`— String karakter untuk dievaluasi.

- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "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"
        }
    }
}
```

HARI INI

Membuat kolom baru yang berisi tanggal saat ini dalam formatyyyy-mm-dd.

Parameter

- `timeZone`— Nama zona waktu. Jika tidak ada zona waktu yang ditentukan, maka defaultnya adalah Universal Coordinated Time (UTC).
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "TODAY",
        "Parameters": {
            "timeZone": "US/Pacific",
            "targetColumn": "TODAY Column 1"
        }
    }
}
```

UNIX_WAKTU

Membuat kolom baru yang berisi angka yang mewakili waktu epoch (waktu Unix) —jumlah detik sejak 1 Januari 1970—berdasarkan kolom sumber atau nilai masukan. Jika zona waktu dapat disimpulkan, output berada di zona waktu tersebut. Jika tidak, outputnya ada di Universal Coordinated Time (UTC).

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME",
        "Parameters": {
            "sourceColumn": "TIME Column 1",
            "targetColumn": "TIME Column 1_UNIXTIME"
        }
    }
}
```

FORMAT UNIX_TIME_

Mengkonversi waktu Unix untuk kolom sumber atau nilai masukan ke format tanggal numerik tertentu, dan mengembalikan hasilnya dalam kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.

- `value`— Sebuah integer yang mewakili stempel waktu Unix epoch.
- `dateTimeFormat` – Opsional. String format untuk tanggal, seperti yang akan muncul di kolom baru. Jika tidak ditentukan, format defaultnya adalah`yyyy-mm-dd HH:MM:SS`.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "UNIX_TIME_FORMAT",
        "Parameters": {
            "value": "1601936554",
            "dateTimeFormat": "yyyy-mm-dd HH:MM:SS",
            "targetColumn": "UNIXTIMEFORMAT Column 1"
        }
    }
}
```

WEEK_DAY

Membuat kolom baru yang berisi hari dalam seminggu, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "WEEK_DAY",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEKDAY"
        }
    }
}
```

WEEK_NUMBER

Membuat kolom baru yang berisi jumlah minggu (dari 1 hingga 52), dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "WEEK_NUMBER",
        "Parameters": {
            "sourceColumn": "DATETIME Column 1",
            "targetColumn": "DATETIME Column 1_WEEK_NUMBER"
        }
    }
}
```

YEAR

Membuat kolom baru yang berisi tahun, dari string yang mewakili tanggal.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

> **Note**
>
> Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "YEAR",
        "Parameters": {
            "value": "2019-06-12",
            "targetColumn": "YEAR Column 1"
        }
    }
}
```

Fungsi jendela

Mengikuti, temukan topik referensi untuk fungsi jendela yang bekerja dengan tindakan resep.

Topik

- FILL
- SELANJUTNYA
- SEBELUMNYA
- ROLLING_AVERAGE
- ROLLING_COUNT_A

- ROLLING_KTH_LARGEST
- ROLLING_KTH_LARGEST_UNIQUE
- ROLLING_MAX
- BERGULING_MIN
- ROLLING_MODE
- ROLLING_STANDARD_DEVIATION
- ROLLING_SUM
- ROLLING_VARIANCE
- ROW_NUMBER
- SESI

FILL

Mengembalikan kolom baru berdasarkan kolom sumber tertentu. Untuk nilai yang hilang atau nol di kolom sumber, `FILL` pilih nilai nonblank terbaru dari jendela baris sebelum dan sesudah nilai sumber yang dimaksud. Nilai yang dipilih kemudian ditempatkan di kolom baru.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "FILL",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "last_name",
            "targetColumn": "last_name_FILL"
        }
    }
```

```
}
```

SELANJUTNYA

Mengembalikan kolom baru, di mana setiap nilai mewakili nilai yang n baris kemudian di kolom sumber.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRows`— Nilai yang mewakili n baris sebelumnya di kolom sumber. Misalnya, jika `numRows` adalah 3, maka `NEXT` gunakan nilai ketiga berikutnya sebagai `sourceColumn` `targetColumn` nilai baru.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "NEXT",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_NEXT"
        }
    }
}
```

SEBELUMNYA

Mengembalikan kolom baru, di mana setiap nilai mewakili nilai yang n baris sebelumnya di kolom sumber.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRows`— Nilai yang mewakili n baris sebelumnya di kolom sumber. Misalnya, jika `numRows` adalah 3, maka `PREV` gunakan nilai ketiga sebelumnya sebagai `sourceColumn` `targetColumn` nilai baru.

- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "PREV",
        "Parameters": {
            "numRows": "1",
            "sourceColumn": "age",
            "targetColumn": "age_PREV"
        }
    }
}
```

ROLLING_AVERAGE

Mengembalikan dalam kolom baru rata-rata bergulir nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_AVERAGE",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_AVERAGE"
        }
```

```
    }
}
```

ROLLING_COUNT_A

Mengembalikan dalam kolom baru jumlah bergulir nilai non-null dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_COUNT_A",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_COUNT_A"
        }
    }
}
```

ROLLING_KTH_LARGEST

Mengembalikan dalam kolom baru nilai bergulir k th terbesar dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.

- `value`Nilai untuk k.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
  "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

Mengembalikan dalam kolom baru bergulir unik k th nilai terbesar dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `value`Nilai untuk k.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
  "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

Mengembalikan dalam kolom baru bergulir maksimum nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_MAX",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MAX"
        }
    }
}
```

BERGULING_MIN

Mengembalikan dalam kolom baru minimum bergulir nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_MIN",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MIN"
        }
    }
}
```

ROLLING_MODE

Mengembalikan dalam kolom baru mode bergulir (nilai paling umum) dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- ModeType — Fungsi modal untuk diterapkan ke jendela. Nilai yang valid adalah `NONE`, `MINIMUM`, `MAXIMUM`, dan `AVERAGE`.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_MODE",
        "Parameters": {
            "modeType": "MINIMUM",
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_MODE"
        }
    }
}
```

ROLLING_STANDARD_DEVIATION

Mengembalikan dalam kolom baru standar deviasi bergulir nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_STDEV",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_STDEV"
        }
    }
}
```

ROLLING_SUM

Mengembalikan dalam kolom baru jumlah bergulir nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom tertentu.

Parameter

- `sourceColumn`— Nama kolom yang ada.

  `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_SUM",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_SUM"
        }
    }
}
```

ROLLING_VARIANCE

Mengembalikan dalam kolom baru varians bergulir nilai dari jumlah tertentu baris sebelum ke jumlah tertentu baris setelah baris saat ini di kolom yang ditentukan.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `numRowsBefore`— Sejumlah baris sebelum baris sumber saat ini, mewakili awal jendela.
- `numRowsAfter`— Sejumlah baris setelah baris sumber saat ini, mewakili akhir jendela.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROLLING_VAR",
        "Parameters": {
            "numRowsAfter": "10",
            "numRowsBefore": "10",
            "sourceColumn": "weight_kg",
            "targetColumn": "weight_kg_ROLLING_VAR"
        }
    }
}
```

ROW_NUMBER

Mengembalikan di kolom baru pengidentifikasi sesi berdasarkan jendela yang dibuat oleh nama kolom dari pernyataan “grup oleh” dan “urutan berdasarkan”.

Parameter

- `groupByColumns`— JSON-encoded String yang menggambarkan kolom “grup dengan”.
- `orderByColumns`— Sebuah JSON-encoded string yang menggambarkan kolom “order by”.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "ROW_NUMBER",
        "Parameters": {
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "Row number"
        }
    }
}
```

SESI

Mengembalikan di kolom baru pengidentifikasi sesi berdasarkan jendela yang dibuat oleh nama kolom dari pernyataan “grup oleh” dan “urutan berdasarkan”.

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `units`— Satuan ukuran untuk menggambarkan panjang sesi. Nilai yang valid adalah `MONTHSYEARS`,`MILLISECONDS`,`QUARTERS`,`HOURS`,`MICROSECONDS`,`WEEKS`,`SECONDS`,`DAYS`, dan`MINUTES`.
- `value`— Jumlah `units` untuk menentukan periode waktu.
- `groupByColumns`— JSON-encoded String yang menggambarkan kolom “grup dengan”.
- `orderByColumns`— Sebuah JSON-encoded string yang menggambarkan kolom “order by”.
- `targetColumn`— Nama untuk kolom yang baru dibuat.

Example Contoh

```
{
    "Action": {
        "Operation": "SESSION",
        "Parameters": {
            "sourceColumn": "object number",
            "units": "MINUTES",
            "value": "10",
            "groupByColumns": "[\"is public domain\"]",
            "orderByColumns": "[\"dimensions\"]",
            "targetColumn": "object number_SESSION",
        }
    }
}
```

Fungsi web

Berikut, temukan topik referensi untuk fungsi web yang bekerja dengan tindakan resep.

Topik

- IP_TO_INT
- INT_TO_IP
- URL_PARAMS

IP_TO_INT

Mengkonversi nilai Internet Protocol versi 4 (IPv4) dari kolom sumber atau nilai lainnya ke nilai integer yang sesuai di kolom target, dan mengembalikan hasilnya di kolom baru. Fungsi ini hanya berfungsi untuk IPv4.

Misalnya, perhatikan alamat IP berikut.

```
192.168.1.1
```

Jika Anda menggunakan nilai ini sebagai input ke`IP_TO_INT`, nilai output adalah sebagai berikut.

```
3232235777
```

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "IP_TO_INT",
        "Parameters": {
            "sourceColumn": "my_ip_address",
            "targetColumn": "IP_TO_INT Column 1"
        }
    }
}
```

INT_TO_IP

Mengkonversi nilai integer dari kolom sumber atau nilai lain ke nilai IPv4 yang sesuai di kemudian kolom target, dan mengembalikan hasilnya dalam kolom baru. Fungsi ini hanya berfungsi untuk IPv4.

Misalnya, perhatikan bilangan bulat berikut.

```
167772410
```

Jika Anda menggunakan nilai ini sebagai input ke`INT_TO_IP`, nilai output adalah sebagai berikut.

```
10.0.0.250
```

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
[  {
    "RecipeAction": {
      "Operation": "INT_TO_IP",
      "Parameters": {
        "sourceColumn": "my_integer",
        "targetColumn": "INT_TO_IP Column 1"
      }
    }
  }
]
```

URL_PARAMS

Mengekstrak parameter kueri dari string URL, memformatnya sebagai objek JSON, dan mengembalikan hasilnya di kolom baru.

Misalnya, perhatikan URL berikut.

```
https://example.com/?firstParam=answer&secondParam=42
```

Jika Anda menggunakan nilai ini sebagai input keURL_PARAMS, nilai output adalah sebagai berikut.

```
{"firstParam": ["answer"], "secondParam": ["42"]}
```

Parameter

- `sourceColumn`— Nama kolom yang ada.
- `value`— String karakter untuk dievaluasi.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Anda dapat menentukan salah satu dari `sourceColumn` atau `value`, bukan keduanya.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "URL_PARAMS",
        "Parameters": {
            "sourceColumn": "my_url",
            "targetColumn": "URL_PARAMS Column 1"
        }
    }
}
```

Fungsi lainnya

Mengikuti, temukan topik referensi untuk fungsi lain yang bekerja dengan tindakan resep.

Topik

- BERSATU
- GET_ACTION_RESULT
- GET_STEP_DATAFRAME

BERSATU

Mengembalikan dalam kolom baru nilai non-null pertama ditemukan dalam array kolom. Urutan kolom yang tercantum dalam fungsi menentukan urutan penelusurannya.

Parameter

- `sourceColumns`— Sebuah JSON-encoded string yang mewakili daftar kolom yang ada.
- `targetColumn`— Nama kolom baru yang akan dibuat.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "COALESCE",
        "Parameters": {
            "sourceColumns": "[\"nation_position\",\"joined\"]",
            "targetColumn": "COALESCE Column 1"
        }
    }
}
```

GET_ACTION_RESULT

Mengambil hasil dari tindakan yang dikirimkan sebelumnya. Hanya untuk digunakan dalam pengalaman interaktif.

Parameter

- `actionId`— ActionId Kembali dalam SendProjectSessionAction tanggapan asli.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "GET_ACTION_RESULT",
        "Parameters": {
            "actionId": "7",
```

```
        }
    }
}
```

GET_STEP_DATAFRAME

Mengambil bingkai data dari langkah dalam resep proyek. Hanya untuk digunakan dalam pengalaman interaktif. Digunakan dengan ViewFrame parameter untuk membuat paginasi di seluruh bingkai data yang besar.

Parameter

- `stepIndex`— Indeks langkah dalam resep proyek untuk mengambil bingkai data.

Example Contoh

```
{
    "RecipeAction": {
        "Operation": "GET_STEP_DATAFRAME",
        "Parameters": {
            "stepIndex": "0"
        }
    }
}
```

Kuota untuk AWS Glue DataBrew

Anda dapat melihat kuota DataBrew layanan Anda di konsol AWS Service Quotas. Anda juga dapat meminta kenaikan kuota, untuk kuota apa pun yang dapat disesuaikan.

Riwayat dokumen untuk AWS Glue DataBrew Panduan Developer

Versi API saat ini: databrew-2017-07-25

Tabel berikut menjelaskan dokumentasi untuk rilis ini AWS Glue DataBrew. Jika Anda ingin diberi tahu saat Panduan AWS Glue DataBrew Pengembang diperbarui, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
`glue:GetCustomEntityType` ditambahkan ke kebijakan AWS terkelola	Izin ini diperlukan untuk menjalankan pekerjaan AWS Glue DataBrew profil dengan PII-identification diaktifkan. Untuk informasi selengkapnya, lihat AWS Glue DataBrew pembaruan kebijakan AWS terkelola.	Maret 20, 2024
Support untuk beberapa algoritma hashing dalam transformasi CRYPTOGRAPHIC_HASH	Anda sekarang dapat menentukan algoritma hashing saat hashing nilai dalam kolom. Untuk informasi lebih lanjut, lihat CRYPTOGRAPHIC_HASH.	11 Agustus 2023
`glue:BatchGetCustomEntityTypes` ditambahkan ke kebijakan AWS terkelola	Izin ini diperlukan untuk menjalankan pekerjaan AWS Glue DataBrew profil dengan PII-identification diaktifkan. Untuk informasi selengkapnya, lihat AWS Glue DataBrew pembaruan kebijakan AWS terkelola.	9 Mei 2022

Dukungan untuk format file Apache ORC	DataBrew sekarang mendukung Apache ORC sebagai format file untuk sumber DataBrew data dan output. Untuk informasi selengkapnya, lihat Jenis file yang didukung untuk sumber data.	31 Maret 2022
Dukungan untuk akses AWS Glue Data Catalog Amazon S3 lintas akun	Sekarang Anda dapat mengakses tabel AWS Glue Data Catalog S3 dari tabel lain Akun AWS jika kebijakan sumber daya yang sesuai dibuat di AWS Glue konsol. Setelah membuat kebijakan , tabel Data Catalog S3 yang relevan dapat dipilih sebagai sumber input saat membuat DataBrew kumpulan data. Untuk informasi selengkap nya, lihat Sambungan yang didukung untuk sumber dan output data.	11 Maret 2022

Support untuk integrasi konsol asli dengan Amazon AppFlow	DataBrew sekarang memiliki integrasi konsol asli dengan Amazon AppFlow. Integrasi ini berarti Anda dapat terhubung ke data dari Salesforce, Zendesk, Slack ServiceNo w, dan aplikasi software-as-a-service (SaaS) lainnya. Anda juga dapat terhubung ke data dari Layanan AWS seperti Amazon S3 dan Amazon Redshift. Untuk informasi selengkapnya, lihat Sambungan yang didukung untuk sumber dan output data.	18 November 2021
Support untuk aturan kualitas data	DataBrew sekarang mendukung pembuatan aturan kualitas data, yang merupakan pemeriksaan validasi yang dapat disesuaikan yang menentukan persyaratan bisnis untuk data tertentu. Untuk informasi selengkapnya, lihat Memvalidasi kualitas data di AWS Glue DataBrew.	18 November 2021

Support untuk pernyataan SQL kustom	DataBrew sekarang mendukung pernyataan SQL khusus untuk mengambil data dari Amazon Redshift dan Snowflake. Dukungan ini berarti Anda dapat menggunakan kueri yang dibuat khusus untuk memilih dan membatasi data yang dikembalikan dari tabel besar. Untuk informasi selengkapnya, lihat Sambungan yang didukung untuk sumber dan output data.	18 November 2021
Support untuk deteksi PII	DataBrew sekarang mendukung deteksi informasi identitas pribadi (PII). Ini memberi Anda opsi untuk menutupi PII selama persiapan data. Untuk informasi selengkapnya, lihat Mengidentifikasi dan menangani informasi identitas pribadi (PII).	18 November 2021
Support untuk AWS Wilayah tambahan	DataBrew sekarang mendukung AWS Wilayah tambahan. Untuk daftar Wilayah yang didukung, lihat AWS Glue DataBrew titik akhir dan kuota.	5 Oktober 2021

Support untuk menulis data ke tabel Lake Formation-based Amazon S3	DataBrew sekarang mendukung penulisan data ke dalam tabel AWS Glue Data Catalog S3 berdasark an AWS Lake Formation . DataBrew juga sekarang mendukung penulisan data ke dalam format Tableau Hyper. Untuk informasi selengkapnya, lihat Membuat dan bekerja dengan pekerjaan AWS Glue DataBrew resep.	13 Agustus 2021
Support untuk menulis data ke destinasi JDBC	DataBrew sekarang mendukung penulisan data langsung ke JDBC-supp orted database dan gudang data. Ini termasuk Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database, dan PostgreSQL. Untuk informasi selengkapnya, lihat Membuat dan bekerja dengan pekerjaan AWS Glue DataBrew resep.	23 Juli 2021
Support untuk menentukan statistik kualitas data yang dihasilkan untuk pekerjaan profil	DataBrew sekarang mendukung menentukan statistik kualitas data mana yang dibuat secara otomatis untuk kumpulan data dalam pekerjaan profil. Untuk informasi selengkapnya, lihat Membuat dan bekerja dengan pekerjaan AWS Glue DataBrew resep.	23 Juli 2021

Support untuk menulis dataset ke dalam AWS Glue Data Catalog	DataBrew sekarang termasuk dukungan untuk menulis kumpulan data langsung ke file.AWS Glue Data Catalog Anda dapat memilih untuk menyimpan kumpulan data yang dibuat dari pekerjaan yang menjalankan resep persiapan data Anda di Amazon S3, Amazon Redshift, dan Amazon RDS tabel di Katalog Data. Tabel RDS yang didukung termasuk untuk Amazon Aurora, RDS untuk Oracle, RDS untuk Microsoft SQL Server, RDS untuk MySQL, dan RDS untuk PostgreSQL.	30 Juni 2021
Support untuk mengidentifikasi tipe data tingkat lanjut	DataBrew sekarang termasuk dukungan untuk secara otomatis mengidentifikasi dan menandai tipe data lanjutan untuk kolom, yang membuatnya lebih mudah untuk menormalkan kolom yang berisi jenis data tertentu. Jenis data ini termasuk nomor Jaminan Sosial, alamat email, nomor telepon, jenis kelamin, kartu kredit, URL, alamat IP, tanggal dan waktu, mata uang, kode pos, negara, wilayah, negara bagian, dan kota.	30 Juni 2021

Support untuk menggunakan Amazon AppFlow untuk mentransfer data dari aplikasi SAAS	DataBrew sekarang mendukung penggunaan Amazon AppFlow untuk mentransfer data ke Amazon S3 dari aplikasi perangkat lunak sebagai layanan (SaaS) pihak ketiga seperti Salesforce, Zendesk, Slack, dan. ServiceNow Untuk informasi selengkapnya, lihat Sambungan yang didukung untuk sumber dan output data.	29 April 2021
Support untuk membuat DataBrew dataset dengan masukan dari database JDBC	DataBrew sekarang mendukung pembuatan kumpulan data dari data dalam JDBC-supported database dan gudang data, termasuk Amazon Redshift, Snowflake, Microsoft SQL Server, MySQL, Oracle Database, dan PostgreSQL. Untuk informasi selengkapnya, lihat Sambungan yang didukung untuk sumber dan output data.	2 April 2021
Support untuk tambahan Wilayah AWS	DataBrew sekarang mendukung tambahan Wilayah AWS. Untuk daftar Wilayah yang didukung, lihat AWS Glue DataBrew titik akhir dan kuota.	28 Januari 2021

Transformasi baru untuk menangani duplikasi	Empat transformasi baru untuk menangani duplikasi telah ditambahkan ke DataBrew konsol dan API. Untuk informasi selengkapnya, lihat DELETE_DUPLICATE_ROWS, FLAG_DUPLICATE_ROWS, FLAG_DUPLICATES_IN_COLUMN, dan REMOVE_DUPLICATES dalam langkah resep kualitas data.	28 Januari 2021
Pembatas CSV tambahan	DataBrew sekarang mendukung pembatas tambahan selain koma dalam file nilai dipisahkan koma (CSV) yang digunakan untuk membuat kumpulan data. DataBrew Untuk informasi selengkapnya, lihat Membuat dan menggunakan AWS Glue DataBrew kumpulan data.	28 Januari 2021
DataBrew ekstensi untuk JupyterLab	Sekarang Anda dapat menggunakan AWS Glue DataBrew sebagai ekstensi di JupyterLab. Untuk informasi selengkapnya, lihat Menggunakan DataBrew sebagai ekstensi di JupyterLab.	20 November 2020
Alat persiapan data baru:AWS Glue DataBrew	Panduan ini adalah perilisan pertama dari Panduan Developer AWS Glue DataBrew.	11 November 2020

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.