



PADDYPOWER.

 betfair

# Crearea obiectelor in BQ



- Proiect nou în Google Cloud Platform
- Dataset nou în BigQuery
  - Tabele noi în dataset -> pe baza de schemă sau completarea fiecărei coloane
- (Optional) Un bucket nou in Google Storage
- Popularea tabelelor cu date de test
  - Mockaroo -> încărcare fișiere csv în Google Storage
  - Faker -> generare date de test cu ajutorul librăriei Faker
  - BigQuery API -> import data cu ajutorul API-ului (Python)

# Crearea obiectelor in BQ



Viewing pinned projects.

music-streaming-332016



Open



Create dataset



## Create dataset

Project ID

music-streaming-332016

CHANGE

Dataset ID \*

star\_schema

Letters, numbers, and underscores allowed

Data location

europe-west6 (Zurich)



### Default table expiration

Enable table expiration

Default maximum table age

Days

### Encryption

Google-managed encryption key

No configuration required

Customer-managed encryption key (CMEK)

Manage via Google Cloud Key Management Service

CREATE DATASET

CANCEL

# Crearea obiectelor in BQ

PADDYPOWER.

betfair

Viewing pinned projects.

music-streaming-332016



star\_schema



nyc-tlc



x-sell-betfair-1



Open

Delete

Create table



Create table



## Source

Create table from  
Empty table

## Destination

Project \* music-streaming-332016 [BROWSE](#)

Dataset ID \* star\_schema

Table name \* dim\_account

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type Native table

## Schema

Edit as text

Field name	Type	Mode	Description
account_id	INTEGER	REQUIRED	{
first_name	STRING	NULLABLE	{
last_name	STRING	NULLABLE	}

CREATE TABLE

CANCEL

# Popularea tabelelor



Optiunea 1: Mockaroo + Încărcare fisier csv in Google Storage

The Mockaroo interface allows you to define the schema for your CSV file. The current schema is as follows:

Field Name	Type	Options
id	Row Number	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X
first_name	First Name	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X
last_name	Last Name	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X
email	Email Address	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X
gender	Gender	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X
ip_address	IP Address v4	blank: 0 % <input type="checkbox"/> $\Sigma$ <input type="checkbox"/> X

Below the schema, there are settings for the data generation:

- # Rows: 1000
- Format: CSV
- Line Ending: Unix (LF)
- Include:  header  BOM

At the bottom, there are buttons for DOWNLOAD DATA, PREVIEW, SAVE THIS SCHEMA, and MORE.

The 'Create table' dialog for Google Storage is shown. The configuration is as follows:

**Source**

- Create table from Google Cloud Storage
- Select file from GCS bucket \* music-streaming/dim\_account.csv  BROWSE

**File format** CSV

Source Data Partitioning

**Destination**

- Project \* music-streaming-332016  BROWSE
- Dataset ID \* star\_schema
- Table name \* dim\_account Unicode letters, marks, numbers, connectors, dashes or spaces allowed.
- Table type Native table

**Schema**

- Auto detect
- Edit as text

# Popularea tabelelor



Optiunea 2: generarea datelor de test cu ajutorul librăriei Faker

În acest caz, trebuie să încarcăm un fișier ajutător în Google Storage care conține librăria și să creăm funcțiile de generare de date.

The screenshot shows the Google Cloud Storage interface. On the left, there's a sidebar with icons for Cloud Storage (selected), Browser (highlighted in blue), Monitoring, and Settings. The main area displays a bucket named "music-streaming". The bucket details are as follows:

Location	Storage class	Public access	Protection
europe-west6 (Zurich)	Standard	Not public	None

Below the bucket details, there are tabs for OBJECTS (which is selected), CONFIGURATION, PERMISSIONS, PROTECTION, and LII. Under the OBJECTS tab, it shows the path "Buckets > music-streaming" and provides options to UPLOAD FILES, UPLOAD FOLDER, CREATE FOLDER, and MANAGE HOLDS. There are also filters for "Filter by name prefix only" and "Filter objects and folders". A table lists the contents of the bucket:

Name	Size	Type
faker.js	52 B	application/x-javascript

# Popularea tabelelor

PADDYPOWER.

betfair

Optiunea 2: generarea datelor de test cu ajutorul librăriei Faker

RUN SAVE SCHEDULE MORE

```
1  /** Generator function */
2  CREATE TEMP FUNCTION dummy_account()
3  RETURNS STRUCT<account_id Int,
4          full_name String,
5          email_address String,
6          date_of_birth Timestamp,
7          gender String,
8          address Struct<street_name String,
9                  street_number Int,
10                 city String
11                >,
12                country String
13              >
14 LANGUAGE js
15 AS """
16 var faker = getFaker()
17 var dummy_account = {};
18
19 dummy_account.account_id = faker.random.number();
20 dummy_account.full_name = faker.name.findName();
21 dummy_account.email_address = faker.internet.email();
22 dummy_account.date_of_birth = faker.date.past();
23 let genders = [ 'F' , 'M' ];
24 dummy_account.gender = faker.random.arrayElement(genders);
25 dummy_account.country = faker.address.country();
26
39    /** Insert 1000 dummy records */
40    INSERT INTO
41      `music-streaming-332016.star_schema.dim_account`
42      (SELECT dummy_account().* from UNNEST(GENERATE_ARRAY(1, 1000)));
43
44    select * from `music-streaming-332016.star_schema.dim_account` ;
45
```

# Generarea datelor de test



## Optiunea 3: BigQuery API (Python)

- Avem nevoie de un Service Account pentru access  
IAM & Admin -> Service Accounts -> Create Service Account
- Detaliile Service Account-ului le păstrăm în proiectul local
- Într-un fișier .env îi atribuim un nume pe care apoi îl putem referenția

A screenshot of a terminal window showing a file named 'credentials.env'. The file contains the following content:

```
credentials.env ×  
bi_resources > data > data_api > credentials.env  
5 GOOGLE_APPLICATION_CREDENTIALS='google-credentials.json'  
6  
7 GOOGLE_DATASET='data_explorer'  
8
```

# Generarea datelor de test



## Opțiunea 3: BigQuery API (Python)

```
from google.cloud import bigquery
from dotenv import load_dotenv
import os

# load global variables
load_dotenv(dotenv_path='credentials.env')

GOOGLE_APPLICATION_CREDENTIALS = os.getenv('GOOGLE_APPLICATION_CREDENTIALS')
GOOGLE_DATASET = os.getenv('GOOGLE_DATASET')

client = bigquery.Client()
dataset_ref = client.dataset(GOOGLE_DATASET)
table_ref = dataset_ref.table(table)
client.load_table_from_dataframe(dataframe, table_ref).result()
```

# Calitatea datelor



Defectele datelor pot fi corectate încă dinainte sau chiar în timpul raportării.

Acest lucru este posibil folosind pași intermediari în procesarea datelor.

- Tool-uri externe - teste, verificări, etc
- Verificări interne - cod, loguri sau metrici

Calitatea datelor este foarte importantă

- Detectarea încercărilor de fraudă
- Detectarea încercărilor de spălare de bani
- Protejarea clienților

# Calitatea datelor



Calitatea datelor are mai multe caracteristici:

- Acuratețe și precizie
- Validitate și legitimitate
- Consistență și fiabilitate
- Relevantă și actualitate
- Completitudine și exhaustivitate
- Disponibilitate și accesibilitate
- Granularitate și unicitate

# Calitatea datelor



Este important să verificăm calitatea datelor cu care vom lucra mai departe. De exemplu, pentru fiecare cont ar trebui să existe o singură intrare în baza de date.

```
SELECT
    COUNT(*),
    full_name,
    email_address
FROM
    `music-streaming-332016.star_schema.dim_account`
GROUP BY 2, 3
HAVING COUNT(*) > 1;
```

# Calitatea datelor



```
SELECT
  COUNT(*),
  song_title
FROM
  `music-streaming-332016.star_schema.dim_song`
GROUP BY 2
HAVING COUNT(*) > 1;
```

Processing location: us-west1

Query results

Query complete (0.3 sec elapsed, 415.9 KB processed)

Job information

f0\_ song\_title

2 delectus quia eos

2 voluptatem excepturi itaque

2 quia et accusantium

2 est et voluptas

2 excepturi a reprehenderit

Același lucru e valabil și pentru celelalte tabele.

Este corectă aceasta verificare?

# Calitatea datelor



```
SELECT
    s.stream_id,
    a.full_name,
    a.country,
    song.song_title,
    d.date
FROM
    `music-streaming-332016.star_schema.fact_streams` s
JOIN
    `music-streaming-332016.star_schema.dim_account` a
ON
    a.account_id = s.account_id
JOIN
    `music-streaming-332016.star_schema.dim_song` song
ON
    CAST (s.song_id AS integer) = song.song_id
JOIN
    `music-streaming-332016.star_schema.dim_date` d
ON
    s.stream_date_id = d.date_id
WHERE
    date = '2019-06-05'
    AND full_name = 'Karli Johnston';
```

Cum putem interpreta acest rezultat?

Row	stream_id	full_name	country	song_title	date
1	51431	Karli Johnston	Romania	distinctio dolore et	2019-06-05
2	51431	Karli Johnston	Romania	distinctio dolore et	2019-06-05

# Calitatea datelor



- Defectele datelor pot fi corectate înainte sau chiar la pasul de raportare
- Acest lucru este posibil folosind Tableau Prep (preprocesare / pregatire)
- Sau folosind direct funcții în BigQuery
- Sau pre-procesând datele înaintea încărcării (ex. Pandas)

Toate resursele prezentate sunt disponibile aici:  
[https://github.com/autrefois/bi\\_resources](https://github.com/autrefois/bi_resources)