

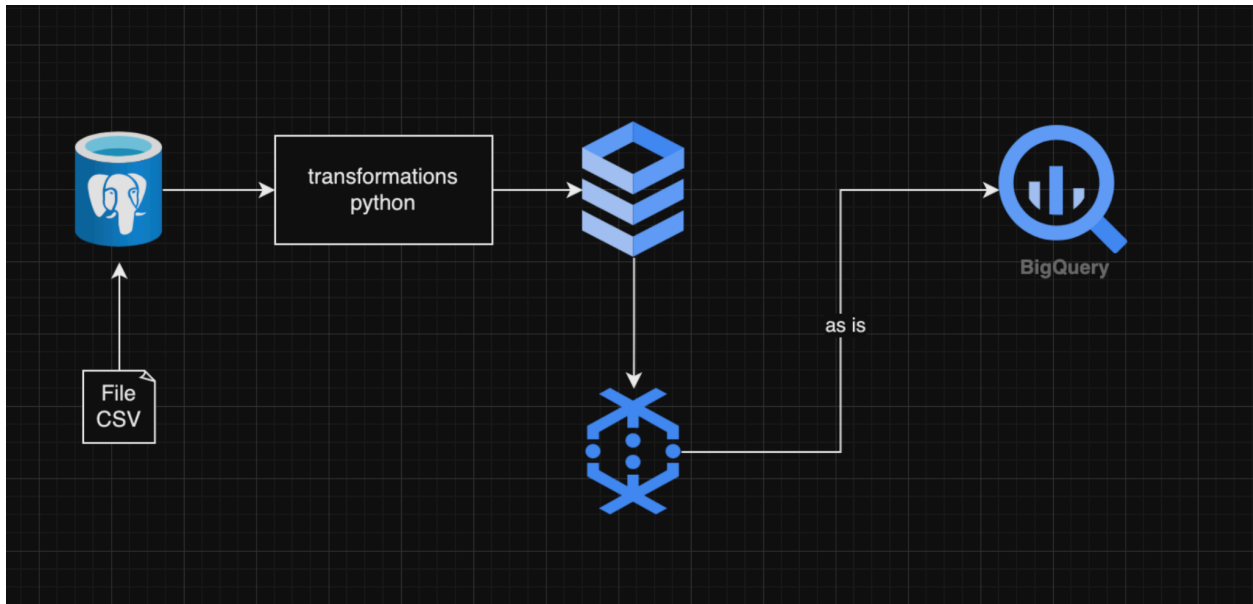
Data Engineer – Data Fellowship 12 IYKRA

Assignment 5

Alya Mutiara Firdausyi

Instruction

Task



1. Buatlah flow seperti pada diagram diatas.
2. Alur flow adalah sebagai berikut:
 - a. Data dari csv pada folder weekly assignment di load pada postgresQL local machine.
 - b. Kemudian lakukan transformasi, filtering, enriching data, dan berikan alasanya mengapa datanya harus dilakukan hal tersebut.
 - c. Setelah datanya bersih, atau sesuai yang dikehendaki, lakukan load ke Cloud SQL di google cloud platform dengan nama table yang sama.
 - d. Setelah datanyanya landing pada Cloud SQL, offload ke BigQuery secara as is atau apa adanya.
3. Di sini bebas akan membuat skenario seperti apa, mau pipelinenya di scheduling boleh tidak juga boleh.
4. Data yang digunakan adalah:
 - a. Dataset banksim artificial → ada di folder

Output

Untuk di week 5 tugasnya sama seperti yang sudah temen temen kerjakan, namun semuanya didokumentasikan melalui github repo transformasi sampai settingan databasenya, data catalog nya, data modelnya. Kemudian invite dan buatlah akun sebagai viewer di gcp temen temen untuk melihat pipeline yang sudah temen temen buat, emailnya adalah sebagai berikut sasongkobgn@gmail.com

Challenge

Goal:

The task involves creating a data pipeline that ingests data from CSV files in a local folder, transforms, filters, and enriches the data, loads the processed data into Cloud SQL, and then offloads the data to BigQuery. The data pipeline can be scheduled or triggered manually.

Step:

1. Data Ingestion

- a. Raw data is collected from Data Fellowship LMS in a CSV format.
- b. A dockerized PostgreSQL is prepared for ingesting the first layer of the database. The latest version from the official docker hub image is used with details as follows:

- i. Host: localhost
- ii. Port: 5432
- iii. Root user: postgres
- iv. Password: postgres

- c. Pull the PostgreSQL image using this command.

```
docker pull postgres:latest
```

Then run the image.

```
docker run --name dockerized-psql -v /Users/alya/Documents/GitHub/DataFellowship12-iykra:/app -p 5432:5432 -e POSTGRES_PASSWORD=postgres -d postgres
```

```
((myenv) alya@192 ~/D/G/DataFellowship12-iykra (main)> docker run --name local-psql -v /Users/alya/Documents/GitHub/DataFellowship12-iykra:/app -p 5432:5432 -e POSTGRES_PASSWORD=postgres -d postgres
58e8747575822941d04c918bd3f3070a98b32a9e18e44e960934ffae50e420ba
((myenv) alya@192 ~/D/G/DataFellowship12-iykra (main)> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                               NAMES
58e874757582   postgres "docker-entrypoint.s..." 6 seconds ago    Up 5 seconds    0.0.0.0:5432->5432/tcp             local-psql
```

To check whether the image is running correctly, run this:

```
docker exec -it dockerized-psql psql -U postgres
```

```
[(myenv) alya@192 ~/D/G/DataFellowship12-iykra (main)> docker exec -it local-psql psql -U postgres
psql (16.2 (Debian 16.2-1.pgdg120+2))
Type "help" for help.

postgres=# \l

```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	ICU Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc	en_US.utf8	en_US.utf8			
template0	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres

```

(3 rows)

postgres=#
```

Create a new database for storing the data.

```
docker exec -it dockerized-psql psql -U postgres -c "CREATE DATABASE banksim_db;"
```

Create a user and grant all access to the user.

```
docker exec -it dockerized-psql psql -U postgres -c "CREATE USER banksim WITH ENCRYPTED PASSWORD 'banksim';"
```

```
docker exec -it dockerized-psql psql -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE banksim_db TO banksim;"
```

```
[(myenv) alya@192 ~/D/G/D/W/Assignment_1 (main)> docker exec -it local-psql psql -U postgres -c "CREATE USER banksim WITH ENCRYPTED PASSWORD 'banksim';"
CREATE ROLE
[(myenv) alya@192 ~/D/G/D/W/Assignment_1 (main)> docker exec -it local-psql psql -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE banksim_db TO banksim;"
GRANT
[(myenv) alya@192 ~/D/G/D/W/Assignment_1 (main)> docker exec -it local-psql psql -U postgres -c "\l"
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	ICU Locale	ICU Rules	Access privileges
banksim_db	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=Tc/postgres + postgres=CTc/postgres+ banksim=CTc/postgres
postgres	postgres	UTF8	libc	en_US.utf8	en_US.utf8			
template0	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres + postgres=CTc/postgres

```

(4 rows)

```

d.

e.

2. Data Transformation and Enrichment

Data transformation was done in Jupyter Notebook using Python's libraries such as pandas and psycopg2.

Please see the jupyter notebook file.

3. Data Loading to Cloud SQL

a. In the PostgreSQL server, create an SQL dump file to be uploaded into the Cloud Storage using this command.

```
docker exec -it dockerized-psql -U banksim -d banksim_db -t customers
--no-owner > customers_dump.sql
```

```
docker exec -it dockerized-psql -U banksim -d banksim_db -t transactions
--no-owner > transactions_dump.sql
```

```
[alya@aaraitum ~/D/G/D/Assignment_5 (main)]> docker exec -it dockerized-psql -U banksim -d banksim_db -t customers --no-owner > customers_dump.sql

What's next?
Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug dockerized-psql
Learn more at https://docs.docker.com/go/debug-cli/
[alya@aaraitum ~/D/G/D/Assignment_5 (main)] [126]> docker exec -it dockerized-psql -U banksim -d banksim_db -t transactions --no-owner > transactions_
dump.sql

What's next?
Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug dockerized-psql
Learn more at https://docs.docker.com/go/debug-cli/
[alya@aaraitum ~/D/G/D/Assignment_5 (main)] [126]> ls
bs140513_032310.csv      bsNET140513_032310.csv  customers_dump.sql      transactions_dump.sql
[alya@aaraitum ~/D/G/D/Assignment_5 (main)]>
```

b. Then upload the dump into a Cloud Storage bucket.

← Bucket details [GO TO PATH](#) [REFRESH](#) [LEARN](#)

banksim-bucket

Location	Storage class	Public access	Protection
asia-southeast2 (Jakarta)	Standard	Not public	Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Folder browser [|<](#)

[banksim-bucket](#) [|](#)

Buckets > banksim-bucket

[UPLOAD FILES](#) [UPLOAD FOLDER](#) [CREATE FOLDER](#) [TRANSFER DATA](#) [MANAGE HOLDS](#) [EDIT RETENTION](#) [DOWNLOAD](#) [DELETE](#)

Filter by name prefix only [Filter](#) Filter objects and folders [Show Live objects only](#) [|](#)

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	
<input type="checkbox"/>	customers_dump.sql	130 B	application/octet-stream	May 13, 2024, 9:57:34 AM	Standard	May 13, 2024, 9:57:34	Download
<input type="checkbox"/>	transactions_dump.sql	130 B	application/octet-stream	May 13, 2024, 9:57:34 AM	Standard	May 13, 2024, 9:57:34	Download

c. Create a Cloud SQL instance.

Google Cloud [df12-alya](#) cloud sql [Search](#) [2](#) [?](#) [|](#)

← Create a PostgreSQL instance

Instance info

Instance ID *

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password * [GENERATE](#)

Set a password for the default admin user "postgres". [Learn more](#)

[PASSWORD POLICY](#)

Database version *

Choose a Cloud SQL edition

A Cloud SQL edition determines foundational characteristics of your instance and cannot be changed later. Choose based on your price and performance needs. [Learn more](#)

☐ Enterprise Plus

- 99.99% availability SLA for eligible instances
- High-performance machines, up to 128 vCPUs
- Up to 35 days point-in-time recovery
- Data cache (optional)

☒ Enterprise

- 99.95% availability SLA for eligible instances
- General purpose machines, up to 96 vCPUs
- Up to 7 days point-in-time recovery

Pricing estimate

\$0.18 per hour (estimated, without discounts)

That's about \$4.40 per day.

Feature usage and traffic costs aren't included in estimate

[SHOW COST BREAKDOWN](#)

Summary

Cloud SQL Edition	Enterprise
Region	asia-southeast2 (Jakarta)
DB Version	PostgreSQL 15
vCPUs	2 vCPU
Memory	8 GB
Data Cache	Disabled
Storage	10 GB
Connections	Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled
Network throughput (MB/s)	500 of 500
Disk throughput (MB/s)	Read: 4.8 of 240.0 Write: 4.8 of 240.0
IOPS	Read: 300 of 15,000 Write: 300 of 15,000

Add assets to banksim

1 Add assets
Add either Storage buckets or BigQuery datasets to this zone

2 Advanced settings (optional)
Change deletion policy and discovery settings for all assets

3 Review assets

Add Assets

New asset

Type *
Storage bucket

Display name
banksim-bucket

ID *
banksim-bucket
Letters, numbers and dashes allowed. Cannot be changed after creation.

Description
Optional 0 / 1024

GCS buckets are required to be configured for access by Dataplex.
Learn more

Bucket *
banksim-bucket BROWSE

To upgrade a bucket, please enable BigQuery Connection API. Upgrade operations will fail without this API

☐ Upgrade to managed

5. Data Offloading to BigQuery

- Create a stream configuration in Datastream and do the prerequisite in the Cloud SQL Studio to create the stream replication and stream publication. This way, Datastream will automatically ingest data to the BigQuery whenever they detect changes in the Cloud SQL database.

Google Cloud
df12-demo
Search (/) for resources, docs, products, and more
Search

Stream details
PAUSE
RESUME
DELETE
TAGS
EDIT
VIEW LOGS

banksim-psql-bq
PostgreSQL / BigQuery

Stream ID
banksim-psql-bq

Source profile
psql-bq

Destination profile
bq-conn-profile

Created
Apr 8, 2024, 11:08:37 PM

Updated
Apr 8, 2024, 11:11:28 PM

OVERVIEW
MONITORING
OBJECTS

Properties

Region
asia-southeast2 (Jakarta)

Labels
No labels set

Objects to include
2 tables

Objects to exclude
None

Backfill mode
Automatic

Destination dataset
Dynamic, based on source schema

Staleness limit
0 seconds

Encryption
Google-managed

Tags
None

- After the stream is up and running, ensure that the dataset has appeared in BigQuery.

Google Cloud df12-demo

Search (/) for resources, docs, products, and more

Search

Explorer

Viewing resources.

SHOW STARRED ONLY

- Queries
- Notebooks
- External connections
- rp_banksimpulic
 - customers
 - transactions

SUMMARY

customers

df12-demo.rp_banksimpulic

Last modified Apr 9, 2024, 12:13:19AM UTC+7

Data asia-southeast2

location

customers

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
step	INTEGER	NULLABLE	-	-	-	-	-
customer	STRING(255)	NULLABLE	-	-	-	-	-
age	STRING(5)	NULLABLE	-	-	-	-	-
gender	STRING(3)	NULLABLE	-	-	-	-	-
zipcodeori	STRING(7)	NULLABLE	-	-	-	-	-
merchant	STRING(255)	NULLABLE	-	-	-	-	-
zipmerchant	STRING(10)	NULLABLE	-	-	-	-	-
category	STRING(255)	NULLABLE	-	-	-	-	-
amount	STRING	NULLABLE	-	-	-	-	-
fraud	BOOLEAN	NULLABLE	-	-	-	-	-
datastream_metadata	RECORD	NULLABLE	-	-	-	-	-

EDIT SCHEMA VIEW ROW ACCESS POLICIES

- c. Finally, the data can be queried to do some analytical things.
6. Documentation and Sharing
- The documentation of this task is placed in [this GitHub repository](#).