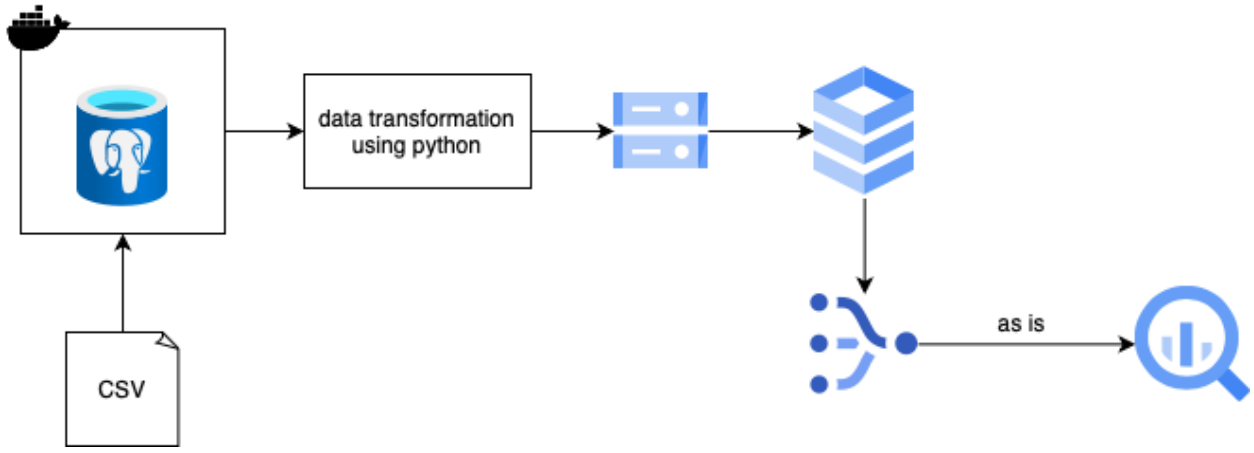
Data Fellowship 12 - Assignment 1

Alya Mutiara Firdausyi

Flow:

1. Load CSV files from local computer to PostgreSQL using Docker.
2. Do data transformation, filtering, enriching and give reasons why the steps must be done.
3. After the data has been cleared out, load it to the Cloud SQL on the Google Cloud Platform with the same table name.
4. After the data exist in the Cloud SQL, pipe the data to BigQuery as it is.

Infrastructure:



Pipeline:

1. Raw data is collected from Data Fellowship LMS in a CSV format.
2. A dockerized PostgreSQL is prepared for ingesting the first layer of the database. The latest version from official docker hub image is used with detail as follows:
    a. Host: localhost
    b. Port: 5432
    c. Root user: postgres
    d. Password: postgres

    Pull 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
                                                  List of databases
   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.

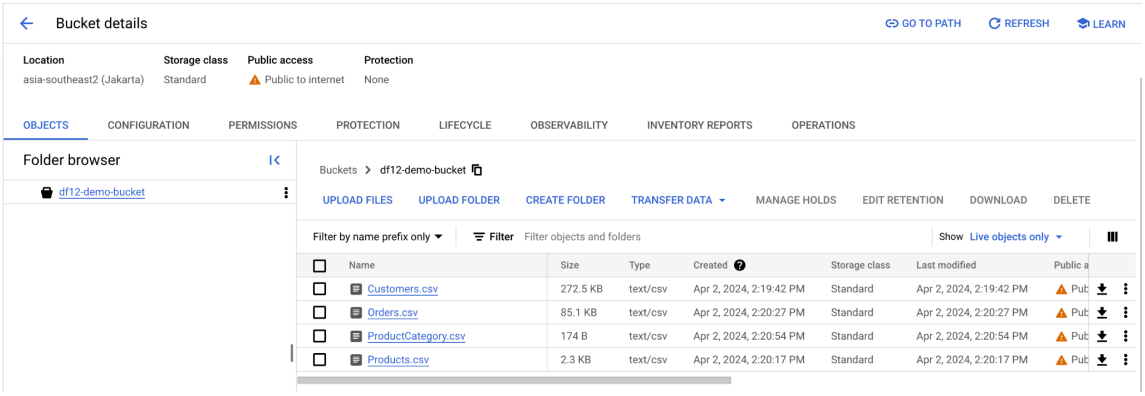
```
docker exec -it dockerized-psql psql -U postgres -c "CREATE USER banksim WITH ENCRYPTED PASSWORD 'banksim';"
docker exec -it dockerized-psql psql -U postgres -c "GRANT ALL PRIVILEGES ON DATABASE banksim_db TO banksim;"
```

```
(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"
                                                  List of databases
    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)
```

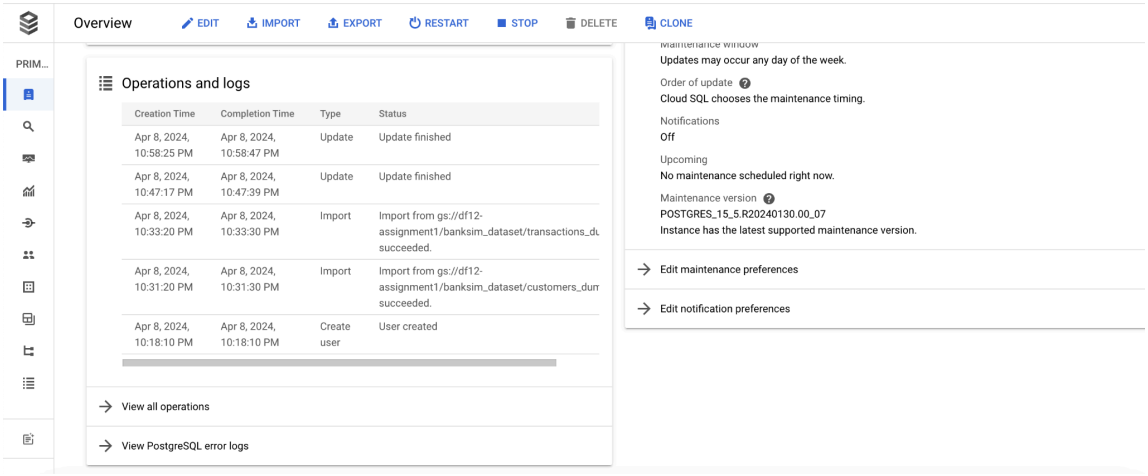
3. Data transformation was done in Jupyter Notebook using Python's library such as pandas and psycopg2.
   Please see the jupyter notebook file.
4. In PostgreSQL server, create SQL dump file to be uploaded into the Cloud Storage using this command.

```
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
```

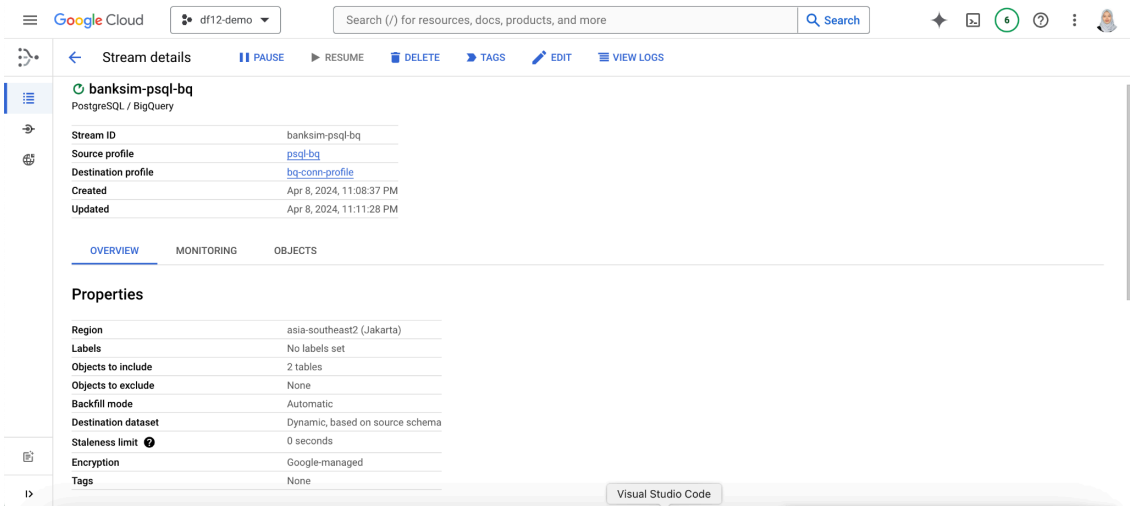
5. Then upload the dump into a Cloud Storage bucket.



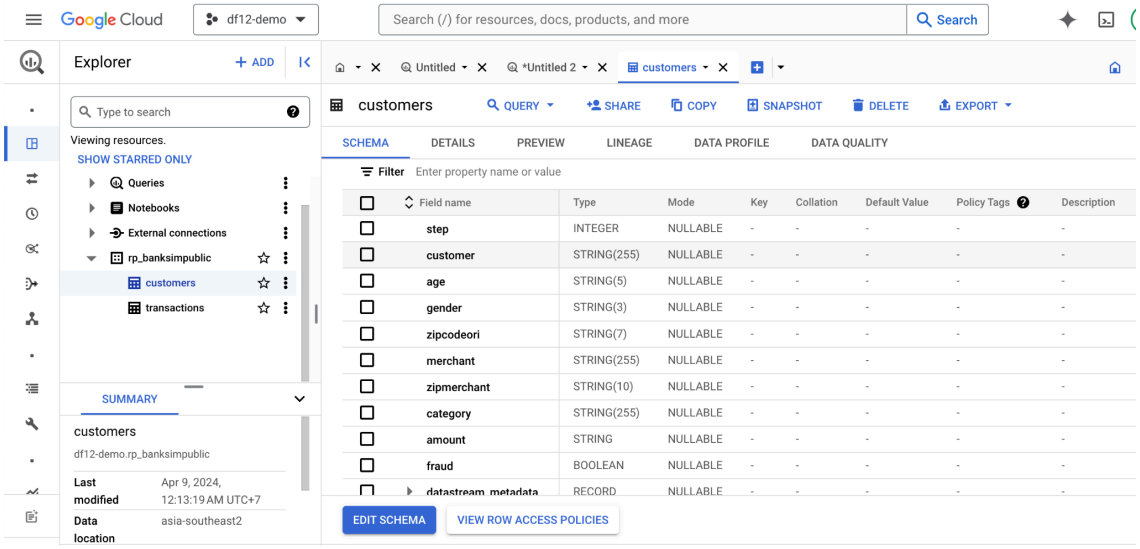
6. Load the data into Cloud SQL by creating database and user beforehand.



7. 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.



8. After the stream is up and running, ensure that the dataset had appeared in BigQuery.



9. Finally, the data can be queried to do some analytical things.