PROJECT ETL

MARÍA FERNANDA TELLO VERGARA

ANA CRISTINA QUINTERO

JOHAN EDELBERTO HURTADO

JAVIER ALEJANDRO VERGARA

ETL

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE

OCTUBRE 04 2024

Context

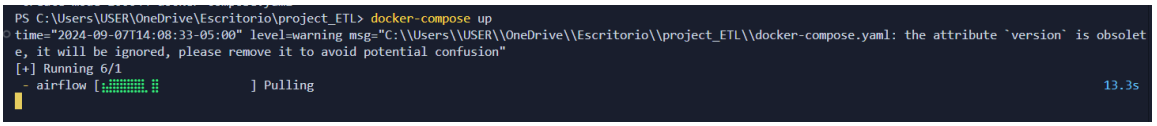
His project focuses on the analysis of a health dataset that includes detailed information on 1,879 patients. Each patient is uniquely identified with an ID between 6000 and 7878.

Tools

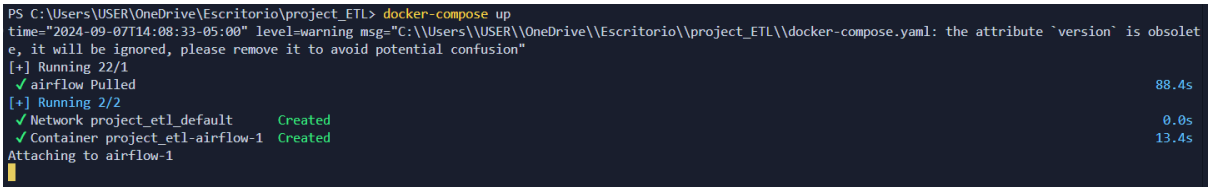
- Python
- Postgress.
- Jupyter Notebook.
- Dataset (Diabetes_healt).
- Git hub.
- Power BI.
- Docker
- AirFlow

Step by step

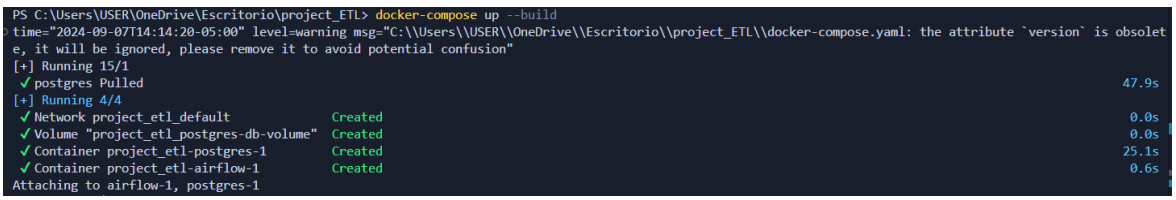
First we download and configure the Airflow image using docker-compose up.



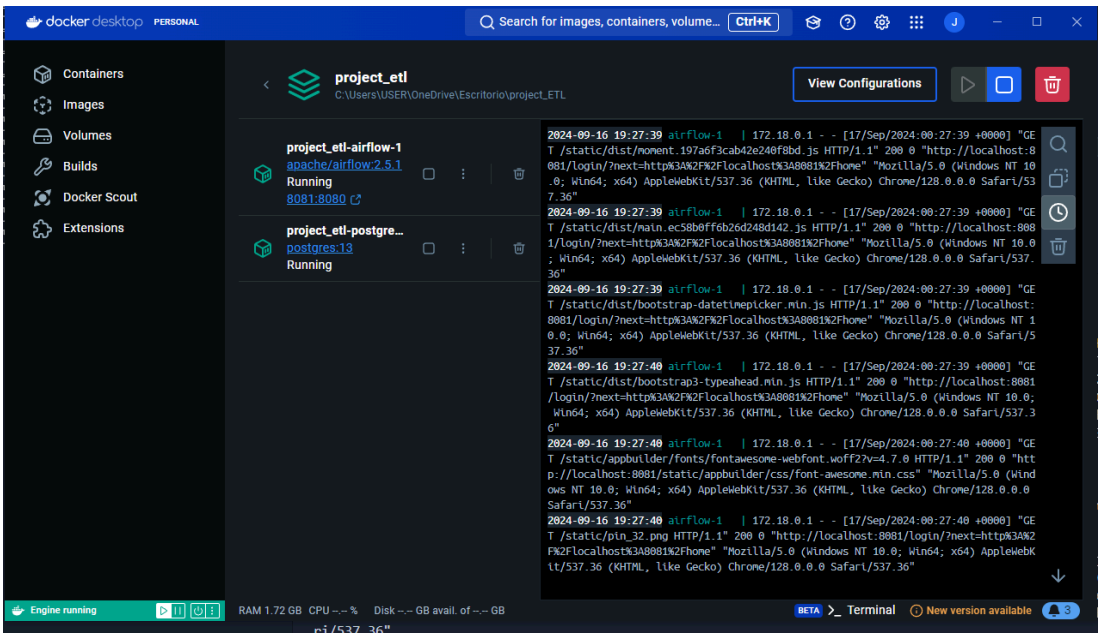
Airflow containers and networks are created and running correctly.



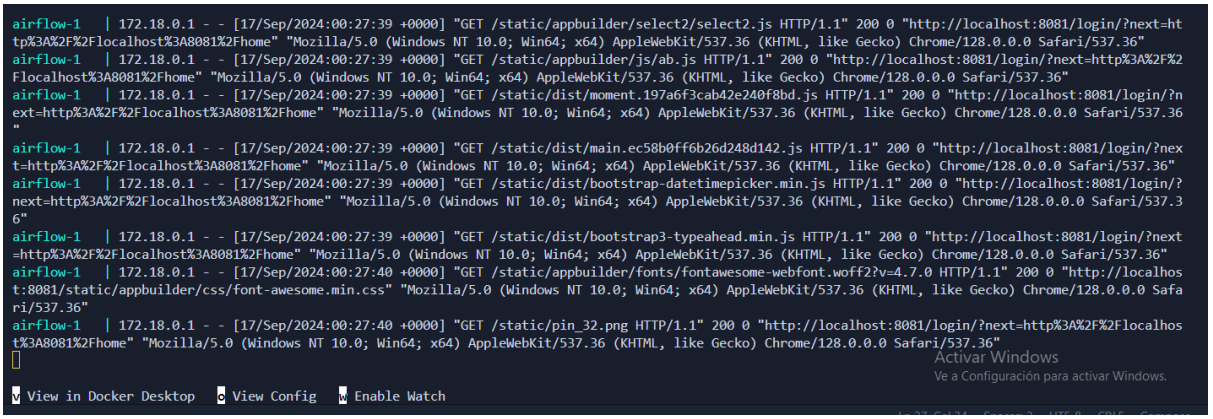
The PostgreSQL and Airflow containers have been started, both are running, and a volume has been created for the PostgreSQL database.

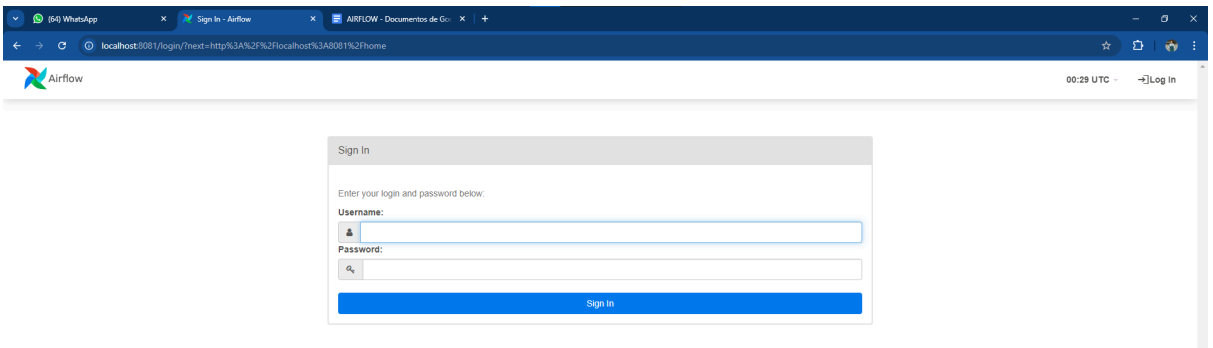


We have Airflow and PostgreSQL containers running successfully in Docker. I can see Airflow logs with activity on localhost:8080.



I can see that Airflow is running and it shows several HTTP activity logs in the container. Additionally, I have access to the Airflow login screen at localhost:8081.

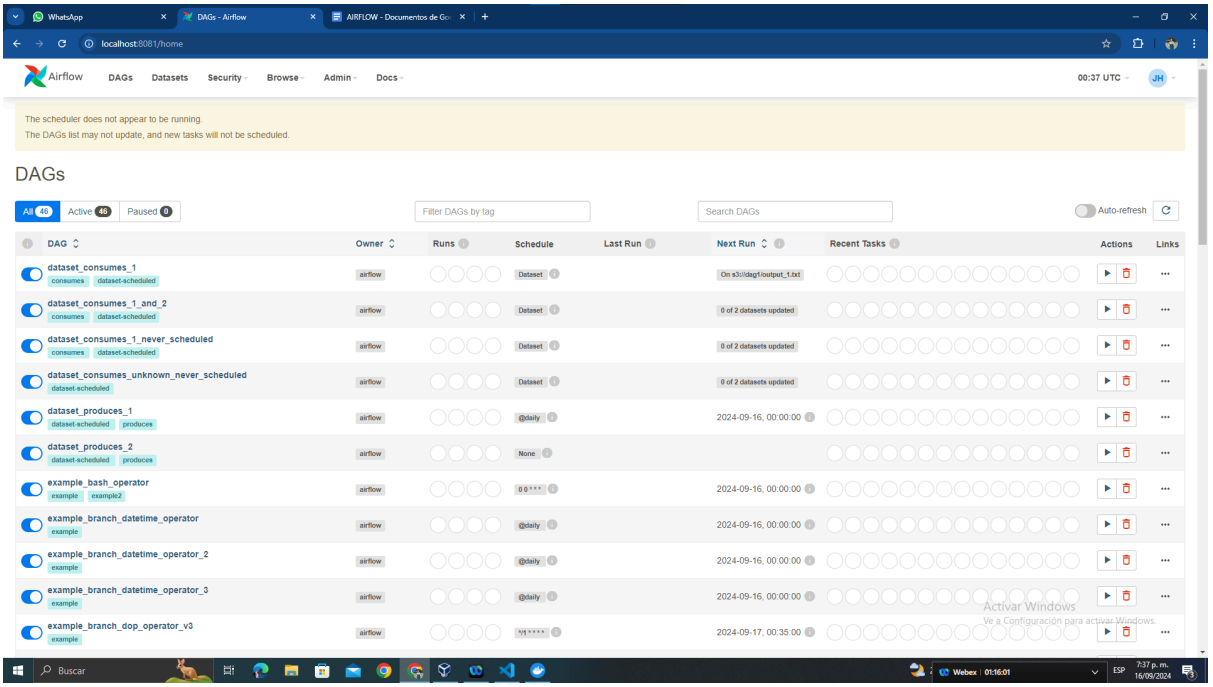




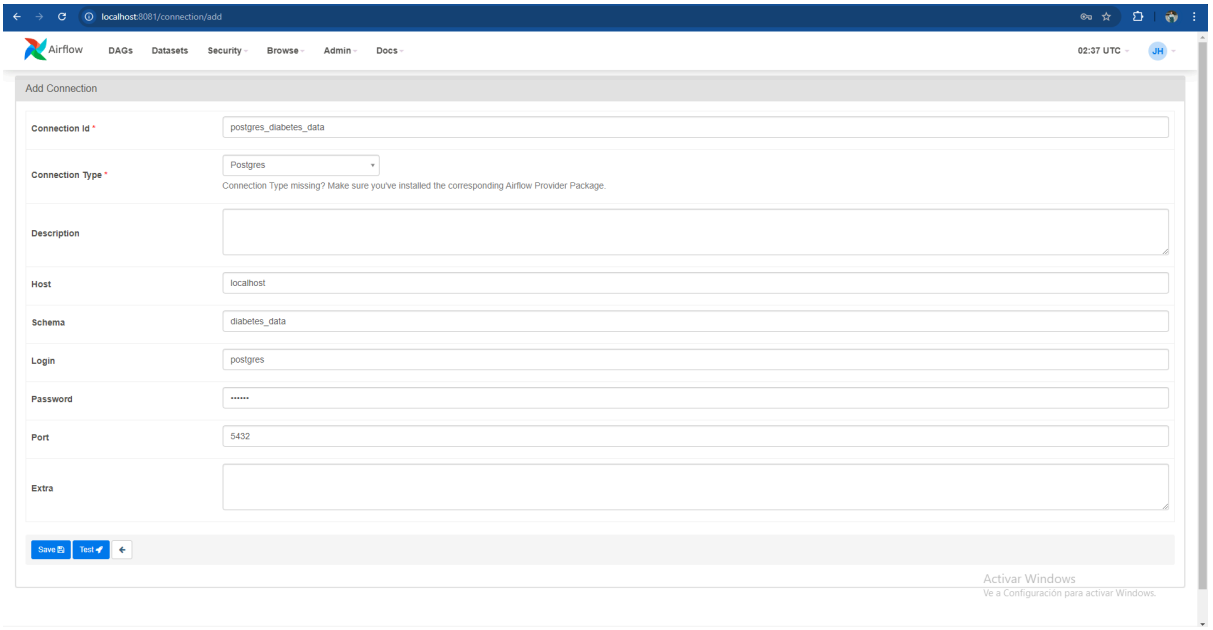
We have created an admin user in Airflow called "admin" with the name Johan Hurtado. The PostgreSQL container is still running, and although we have seen deprecation warnings in the SQLAlchemy configuration, the user was successfully created with the "Admin" role.

```
PS C:\Users\USER\OneDrive\Escritorio\project_ETL> docker-compose run airflow airflow users create --username admin --firstname Johan --lastname Hurtado --role
Admin --email johanhurtaalt@gmail.com --password leganes123
time="2024-09-16T19:35:35-05:00" level=warning msg="C:\\Users\\USER\\OneDrive\\Escritorio\\project_ETL\\docker-compose.yaml: the attribute `version` is obsolet
e, it will be ignored, please remove it to avoid potential confusion"
time="2024-09-16T19:35:35-05:00" level=warning msg="Found orphan containers ([project_etl-airflow-run-c00aaacd9b61 project_etl-airflow-run-a581e8d3c859 project
_etl-airflow-run-9c3b0ea2f677]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orpha
ns flag to clean it up."
[+] Creating 1/0
 ✓ Container project_etl-postgres-1  Running                                                                                                    0.0s

/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:440: DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the
sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
  is_sqlite = "sqlite" in self.get("database", "sql_alchemy_conn")
/home/airflow/.local/lib/python3.7/site-packages/airflow/configuration.py:360: DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the
sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
  self._upgrade_postgres_metastore_conn()
/home/airflow/.local/lib/python3.7/site-packages/airflow/settings.py:249 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_al
chemy_conn option in [database] - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.7/site-packages/airflow/models/base.py:70 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_
alchemy_conn option in [database] - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.7/site-packages/airflow/utils/sqlalchemy.py:47 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the
 sql_alchemy_conn option in [database] - the old setting has been used, but please update your config.
/home/airflow/.local/lib/python3.7/site-packages/airflow/www/app.py:85 DeprecationWarning: The sql_alchemy_conn option in [core] has been moved to the sql_alch
emy_conn option in [database] - the old setting has been used, but please update your config.
[2024-09-17 00:35:42,379] {manager.py:824} WARNING - No user yet created, use flask fab command to do it.
[2024-09-17 00:35:47,073] {providers_manager.py:238} INFO - Optional provider feature disabled when importing 'airflow.providers.google.leveldb.hooks.leveldb.L
evelDBHook' from 'apache-airflow-providers-google' package
[2024-09-17 00:35:48,458] {providers_manager.py:238} INFO - Optional provider feature disabled when importing 'airflow.providers.google.leveldb.hooks.leveldb.L
evelDBHook' from 'apache-airflow-providers-google' package
[2024-09-17 00:35:49,617] {manager.py:212} INFO - Added user admin
User "admin" created with role "Admin"
```

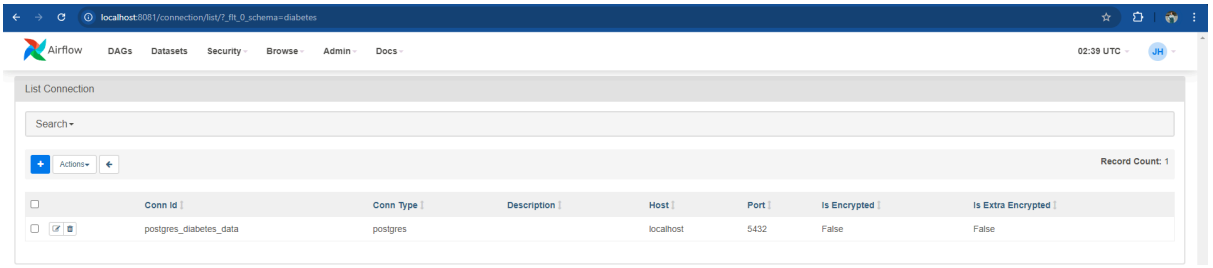
We have accessed the Airflow interface, where several DAGs are displayed ready to run.



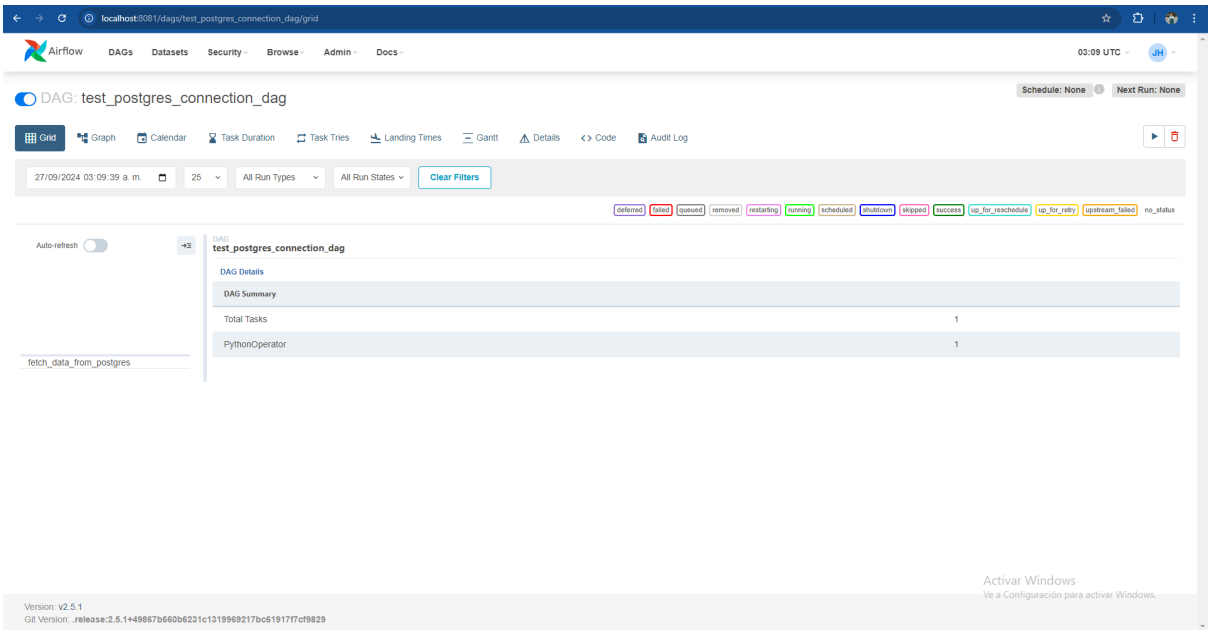
We have set up a connection in Airflow to connect to the PostgreSQL database. The connection type is "Postgres", with the host set to localhost, the schema to clients_data, and the port 5432. The username and password have also been set to allow proper access.



We have created and verified the connection to the diabetes_data database in Airflow, using PostgreSQL. Additionally, we have configured a DAG called test_postgres_connection_dag that performs operations with the connected database, displaying the data extraction task (fetch_data_from_postgres).



We have configured the test_postgres_connection_dag DAG to interact with the diabetes_data database in Airflow.



We entered our PostgreSQL container in Docker using the docker exec -it project_etl-postgres-1 bash command, which allowed us to access the container console to perform tasks directly in PostgreSQL.

```
PS C:\Users\USER> docker exec -it project_etl-postgres-1 bash
root@9ebb5ed9872e:/#
```

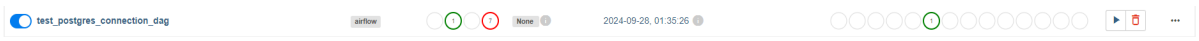
Hemos copiado el archivo de dataset diabetes_data.csv desde Windows al contenedor de Docker utilizando el comando docker cp.

```
C:\Windows\system32\cmd.exe - docker  exec -it project_etl-postgres-1 bash
Microsoft Windows [Versión 10.0.19045.4894]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USER>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS         PORTS                    N
AMES
6da73b17d595   apache/airflow:2.5.1   "/usr/bin/dumb-init …"   22 hours ago   Up 10 minutes  0.0.0.0:8081->8080/tcp   p
roject_etl-airflow-webserver-1
19f7c631c6a3   apache/airflow:2.5.1   "/usr/bin/dumb-init …"   22 hours ago   Up 10 minutes  8080/tcp                 p
roject_etl-airflow-scheduler-1
58ebbf9111a7   postgres:13            "docker-entrypoint.s…"   22 hours ago   Up 10 minutes  5432/tcp                 p
roject_etl-postgres-1

C:\Users\USER>docker cp "C:\Users\USER\OneDrive\Escritorio\project_ETL\diabetes_data.csv" project_etl-postgres-1:/tmp/di
abetes_data.csv
Successfully copied 762kB to project_etl-postgres-1:/tmp/diabetes_data.csv

C:\Users\USER>docker exec -it project_etl-postgres-1 bash
root@58ebbf9111a7:/# ls /tmp
diabetes_data.csv
root@58ebbf9111a7:/#
```

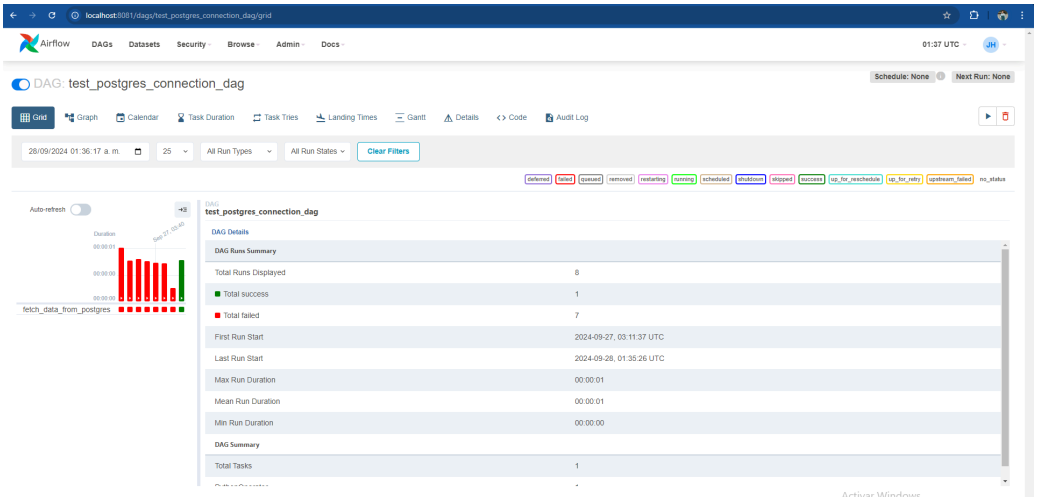
Copied

```
diabetes_data=# COPY diabetes_data FROM '/tmp/diabetes_data.csv' DELIMITER ',' CSV HEADER;
COPY 1879
```

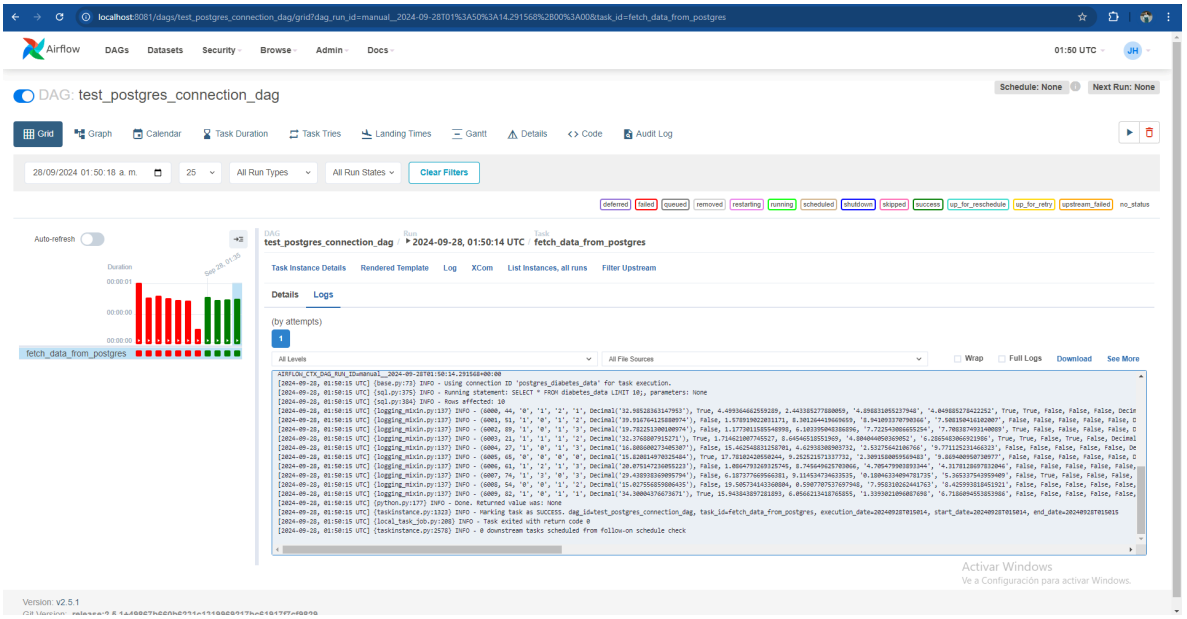
Dag in Airflow



After 8 failed tests, on the ninth run we were able to successfully complete the task in the test_postgres_connection_dag DAG. The error was related to the PostgreSQL connection, as we had not copied the database to the Docker container correctly. After fixing this issue, the DAG runs without any problems.



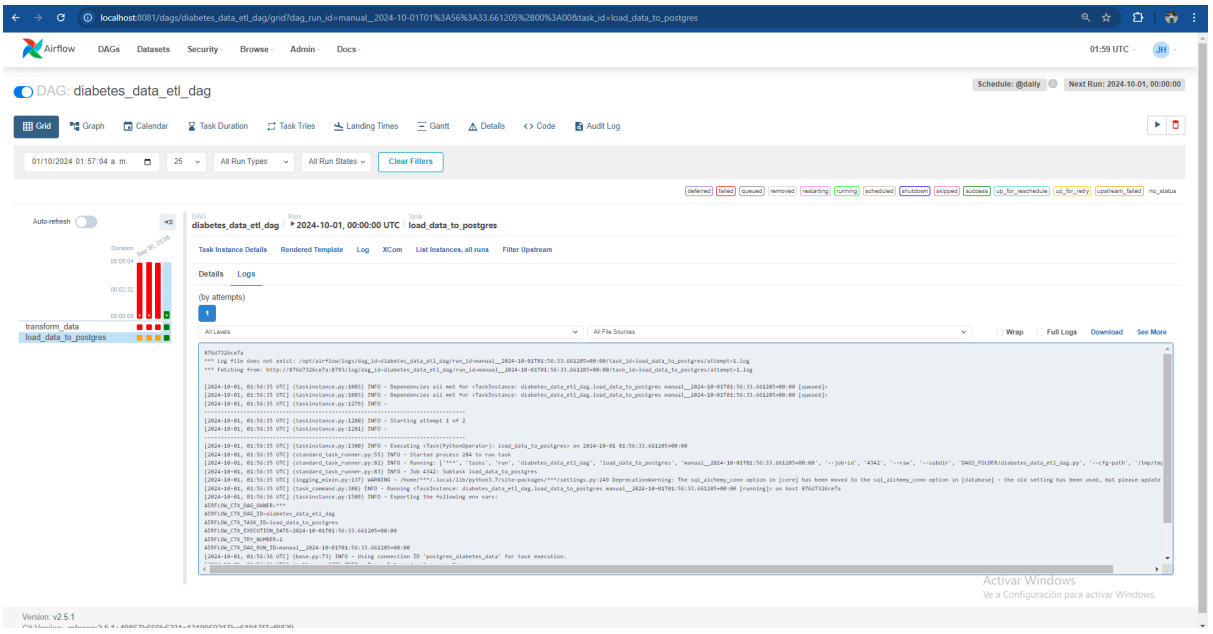
Now we have a problem with the logs, we needed to put the unique key in the docker compose.yml. actually our db in airflow.



CREATION OF THE DIMENSIONAL MODEL

DATA GROUPING AND TRANSFORMATION

After 3 failures the 2 transformation and loading tasks were successful.



We have successfully queried the diabetes_data database tables in PostgreSQL. In the first query, we have obtained the first 5 records from the dim_health_habits table, which shows information about health habits such as smoking, alcohol consumption, physical activity, diet quality, and sleep quality.

```
diabetes_data=# SELECT * FROM dim_health_habits LIMIT 5;
 patientid | smoking | alcoholconsumption | physicalactivity  |    dietquality    |   sleepquality
-----------+---------+--------------------+-------------------+-------------------+-------------------
 6000      | true    | 4.499364662559289  | 2.443385277880059 | 4.898831055237948 | 4.049885278422252
 6001      | false   | 1.578919022031171  | 8.301264419669659 | 8.941093370790366 | 7.508150416102007
 6002      | false   | 1.1773011585548998 | 6.103395048386896 | 7.722543086655254 | 7.708387493140089
 6003      | true    | 1.714621007745527  | 8.64546518551969  | 4.804044050369052 | 6.286548306692199
 6004      | false   | 15.4625488312587   | 4.62938308903732  | 2.53275642106766  | 9.771125231466325
(5 rows)
```

In the second query, we have extracted the first 5 records from the dim_demography table, which contains demographic data such as age, gender, ethnicity, socioeconomic status, and education level.

```
diabetes_data=# SELECT * FROM dim_demography LIMIT 5;
 patientid | age | gender | ethnicity | socioeconomicstatus | educationlevel
-----------+-----+--------+-----------+---------------------+----------------
 6000      |  44 | 0      | 1         | 2                   | 1
 6001      |  51 | 1      | 0         | 1                   | 2
 6002      |  89 | 1      | 0         | 1                   | 3
 6003      |  21 | 1      | 1         | 1                   | 2
 6004      |  27 | 1      | 0         | 1                   | 3
(5 rows)
```

The COPY command in PostgreSQL is used to export the fact_table and dim_health_habits tables to CSV files in the /tmp/ path. The tables are exported in CSV format with the header included.

```
root@9ebb5ed9872e:/# psql -U airflow -d diabetes_data -c "\COPY fact_table TO '/tmp/fact_table.csv' WITH (FORMAT CSV, HEADER)"
COPY 1879
root@9ebb5ed9872e:/# psql -U airflow -d diabetes_data -c "\COPY fact_table TO '/tmp/fact_table.csv' WITH (FORMAT CSV, HEADER)"
COPY 1879
root@9ebb5ed9872e:/# psql -U airflow -d diabetes_data -c "\COPY dim_health_habits TO '/tmp/dim_health_habits.csv' WITH (FORMAT CSV, HEADER)"
COPY 1879
root@9ebb5ed9872e:/#
```

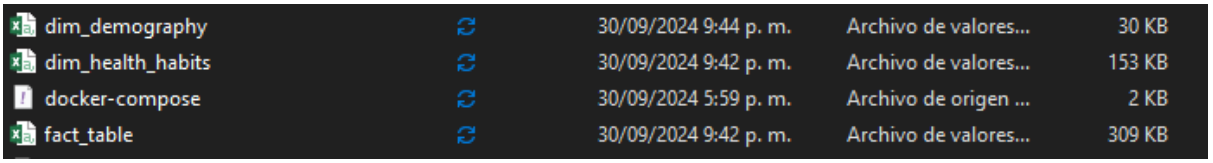
It is verified that the three CSV files (diabetes_data.csv, dim_demography.csv, dim_health_habits.csv) are correctly created in the /tmp/ directory inside the Docker container.

```
root@9ebb5ed9872e:/# ls /tmp
diabetes_data.csv  dim_demography.csv  dim_health_habits.csv  fact_table.csv
root@9ebb5ed9872e:/#
```

Usando el comando docker cp, los archivos CSV se copian desde el contenedor de Docker a una carpeta en el sistema de archivos de Windows, específicamente a la ruta C:\Users\USER\OneDrive\Escritorio\project_ETL.

```
PS C:\Users\USER> docker cp project_etl-postgres-1:/tmp/dim_demography.csv C:\Users\USER\OneDrive\Escritorio\project_ETL
Successfully copied 31.7kB to C:\Users\USER\OneDrive\Escritorio\project_ETL
PS C:\Users\USER> docker cp project_etl-postgres-1:/tmp/fact_table.csv C:\Users\USER\OneDrive\Escritorio\project_ETL
Successfully copied 318kB to C:\Users\USER\OneDrive\Escritorio\project_ETL
PS C:\Users\USER> docker cp project_etl-postgres-1:/tmp/dim_health_habits.csv C:\Users\USER\OneDrive\Escritorio\project_ETL
Successfully copied 158kB to C:\Users\USER\OneDrive\Escritorio\project_ETL
PS C:\Users\USER>
```

We check folders.



we can see that it is fine.

	A
1	patientid,age,gender,ethnicity,socioeconomicstatus,educationlevel
2	6000,44,0,1,2,1
3	6001,51,1,0,1,2
4	6002,89,1,0,1,3
5	6003,21,1,1,1,2
6	6004,27,1,0,1,3
7	6005,65,0,0,0,0
8	6006,61,1,2,1,3
9	6007,74,1,3,0,3
10	6008,54,0,0,1,2
11	6009,82,1,0,1,1
12	6010,59,1,0,2,2
13	6011,82,1,1,1,3
14	6012,79,1,1,0,2
15	6013,22,0,3,0,2
16	6014,29,0,0,1,2
17	6015,76,0,0,1,3
18	6016,80,0,1,1,2

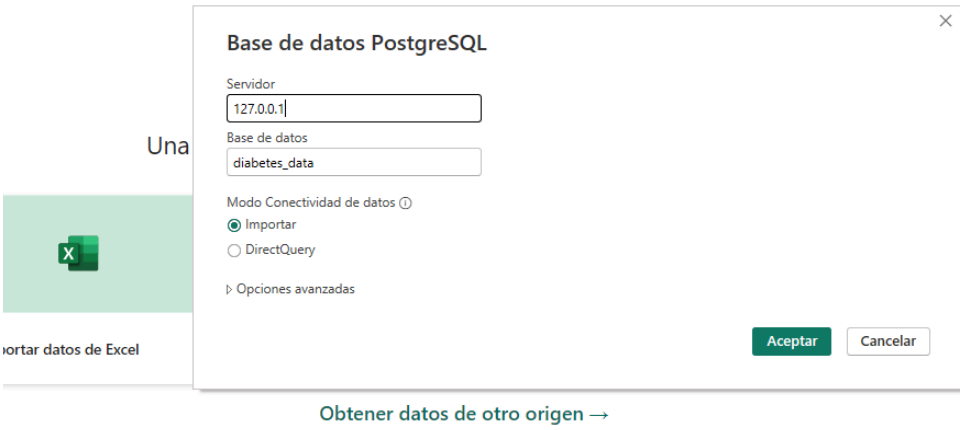
Dimensional modeling is a design technique used in data warehousing and business intelligence systems. It is organized into two types of tables:

- Fact Table: Contains quantitative data, such as health metrics (bmi, systolicbp, diagnosis) in the fact_table table.
- Dimension Tables: Provide context to the facts with descriptive information. In this case, there are the dim_demography (demographic data) and dim_health_habits (health habits) tables.

Reasons for separating the original dataset:

- Efficiency: Improves organization and facilitates quick queries.
- Analysis: Allows data to be analyzed from various perspectives.
- Reduction of redundancy: Minimizes data duplication, improving space and consistency.

BD CONNECTION TO POWER BI



Here are some of the questions we'll be answering in the Power BI dashboard:

- What is the distribution of patients by gender?
- How many patients smoke and how many do not smoke?
- How are patients distributed according to their socioeconomic status?
- How are patient health metrics distributed, including BMI, blood pressure (diastolic and systolic), LDL cholesterol, triglycerides, and quality of life?
- What are the leading causes of death among patients and how many deaths are attributed to each cause?

Finally, we uploaded all the folders to the github repository!