Workshop 2

Presented by:
Juan Camilo Buitrago Gonzalez

ETL

Teacher:
Javier Alejandro Vergara Zorrilla

Universidad Autonoma de Occidente

22/04/2024

Introduction:

This workshop is an exercise on how to build an ETL pipeline using Apache Airflow, the idea is to extract information using two different data sources (csv file, database), then do some transformations and merge the transformed data to finally load into google drive as a CSV file and store the data in a DB. As a last step, create a dashboard from the data stored in the DB to visualize the information.

Technologies used:

The technologies used for this workshop were:

Airflow: is an open-source platform for developing, scheduling, and monitoring batch used for the workflow.

Docker: Used to deploy the airflow application.

Python: The language used for the workshop

Jupyter notebook: The notebook platform used to make the EDA.

Visual Studio Code: The chosen code editor for the workshop management and development

PostgreSQL: The database management system used for storing the spotify and

grammys data

Power BI: The visualization platform used to make the dashboard.

Architecture:

My workshop architecture is organized in a simple structure:

- config: This folder contains all the json files with passwords for privacity.
    - Config_EDA is the file that contains the credentials of the database for localhost, this was created because I could not use the same credentials that in docker.
    - Config contains the credentials for work in the database connection in the dag and the etl process. This is different from config_EDA because this contains a different host.
    - Service_account contains the credentials for work in the store in the google drive.
- dags: This folder contains the dag I used to work in this workshop:

  - Etl file contains all the functions defined of the etl process, like the extraction, the transform, and the upload.
  - Dag file contains the workflow defined.
- Data: This folder contains the two datasets used in the workshop
- Docs: This folder contains the documentation and the dashboard for the workshop
- Notebooks: This folder contains the jupyter notebooks where I made the EDA for both datasets:
  - EDA_001 contains the Exploratory Data Analysis for the Grammys dataset.
  - EDA_002 contains the Exploratory Data Analysis for the Spotify dataset.

- .gitignore: is the file with the names of the ignored files (files that are not committed to the github repository).
- README.md: This file contains the explanation of how to install this program and how it works.
- Dockerfile: Docker builds images automatically by reading the instructions from a Dockerfile which is a text file that contains all commands, in order, needed to build a given image.
- Docker-compose.yaml: used exclusively for local application set-up.
- Requirements.txt: contains the libraries necessary to run the workshop.

Implementation:

Firstable you should create a database in postgresql

Then you should create the files in the folder config with the credentials of your database, its important to know that are necessary to create a config.json for the docker functionality (if you want to insert and do the process in your localhost you should put in you host: host.docker.internal, and the other credentials like user, dbname, password, etc.)

You need to connect to drive using Google Cloud and create a service account to upload it, its important to get permissions and share the folder where you want to store it with the mail of the service account.

Then the process begins by install Docker then run ‘docker-compose up’

and activate the dag called: data_merging_etl_dag.

Then you should run the dag or it will automatically run daily.

Data information

I have 4810 row and 6 columns in Grammys dataset after the transformations. The fields we will use are:

- Title
- Category
- Nominee
- Artist
- Worker
- Winner

I have 89665 rows and 17 columns in Spotify dataset after the transformations. The fields we will use are:

- Artists
- Album_name
- Track_name
- Popularity
- Duration_ms
- Explicit
- Danceability
- Energy
- Loudness
- Speechiness
- Acousticness
- Tempo
- Track_genre
- Genre_category
- Num_artists
- Second_artist
- Popularity_category

For the merge I use the Nominee from Grammys dataset and track_name from spotify dataset.

And I have in my merge 90261 rows and 21 columns, this after delete the column Artist that comes from Grammys dataframe, this because I will not use this column after the merge.

Exploratory Data Analysis (EDA)

During the Exploratory Data Analysis (EDA) process, I encountered several issues with the datasets provided from both the Grammy and Spotify sources that needed addressing to improve data quality and relevance for further analysis.

Grammys Dataset:

The grammys dataset included several columns that were found to contain incorrect or unnecessary information which would not be useful for our analysis objectives. Specifically, columns like img, year, updated_at, and published_at were identified as either irrelevant to the analysis goals or not maintained with reliable data. These columns were subsequently considered for removal or cleaning in the data preprocessing phase.

Spotify Dataset:

Similarly, in the spotify dataset, I noticed several columns that did not contribute valuable information regarding the songs or the artists, which could potentially clutter the dataset. The following columns were identified for exclusion:

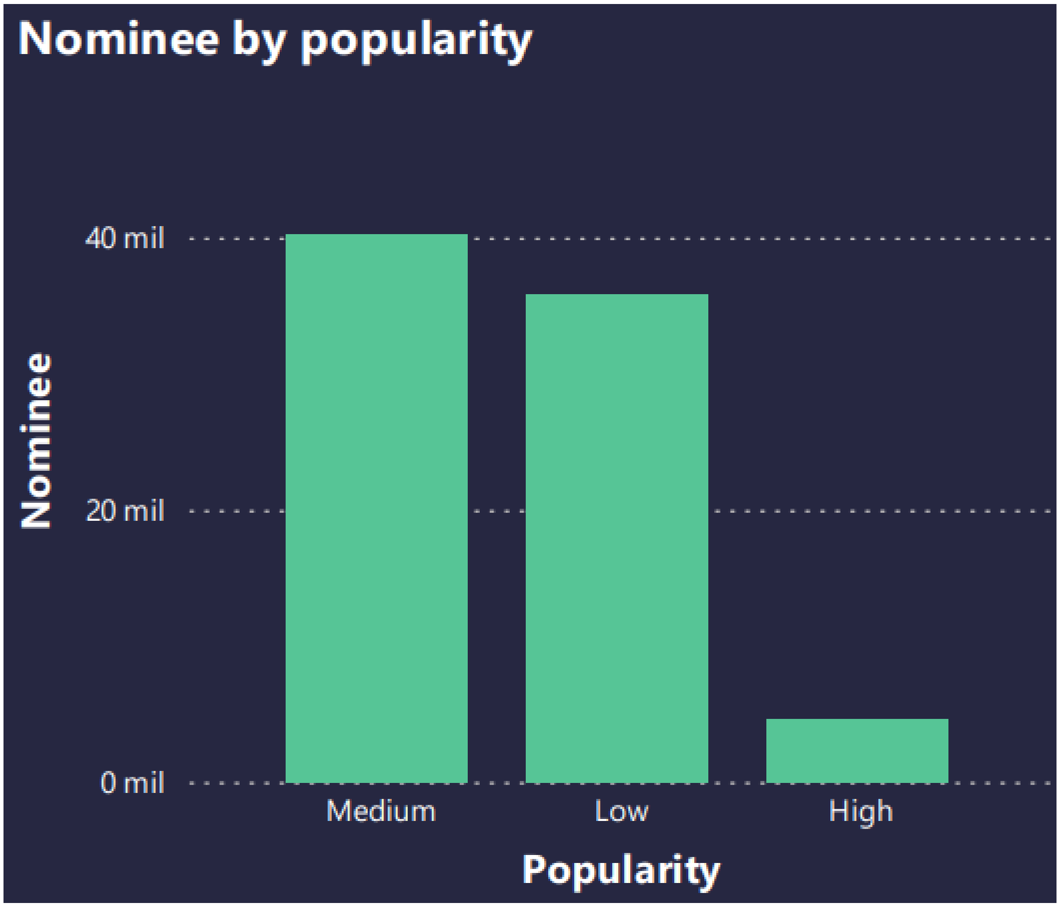
Unnamed: 0: Likely an artifact from data import, serving no analytical purpose.

track_id, key, mode, instrumentalness, time_signature, liveness, valence: These columns, while potentially useful for detailed musicology studies, were not necessary for our high-level analysis and were thus removed to streamline the dataset.

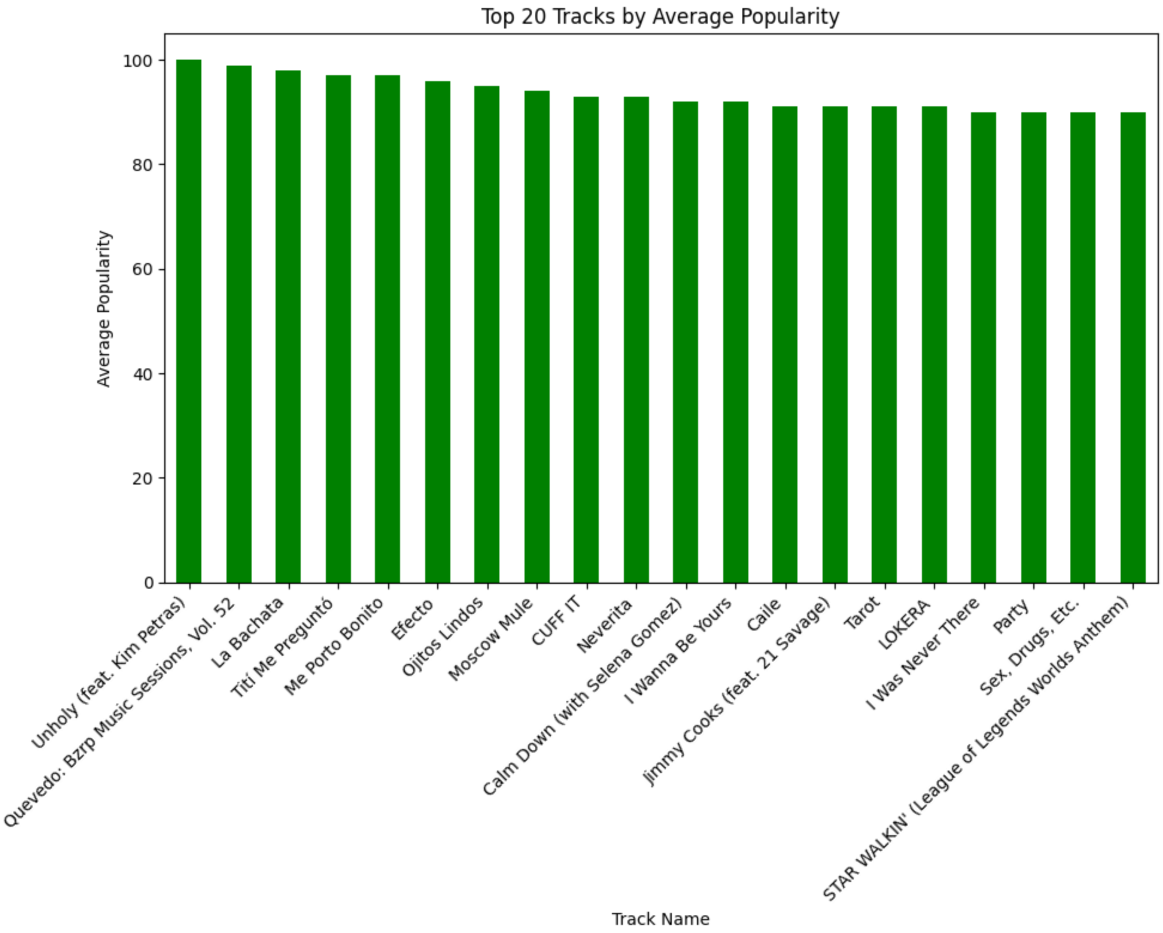
Additionally, the artists column often contained multiple artists per song, which could complicate any analysis involving artist-specific trends or insights. To address this, I split this column to create a new second_artist column, ensuring that each song's additional artist data could be separately and effectively analyzed.

The track_genre column contained a vast array of genres, which was overly granular for our purposes. To manage this, I categorized these genres into broader, more analytically useful categories, allowing for more straightforward aggregation and analysis of genre-related trends in the music data.

These transformations were essential to tailor the datasets for more effective analysis, focusing on relevant data and ensuring the structure supported our analytical goals. This process of refinement helps in minimizing complexities and enhances the clarity of insights derived from the data.

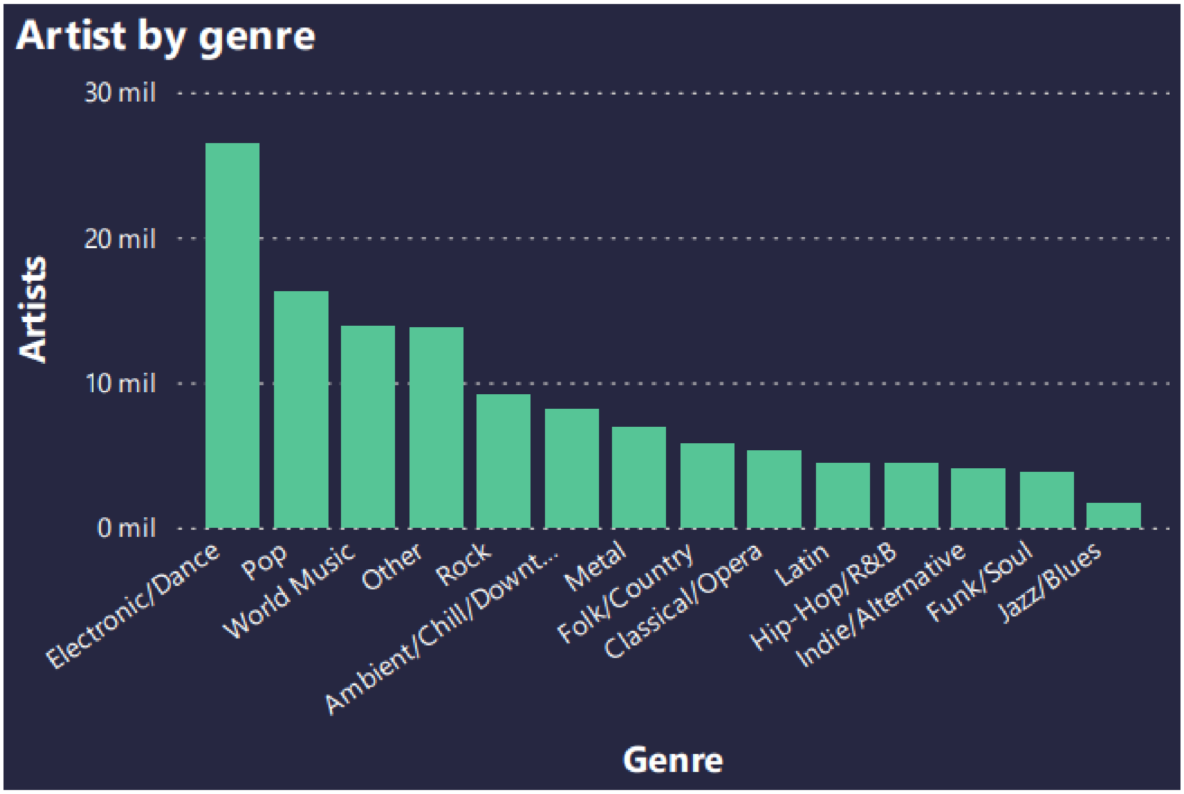


With this graph we can conclude that Is not necessarily to be a very popular song to be a grammy's winner.

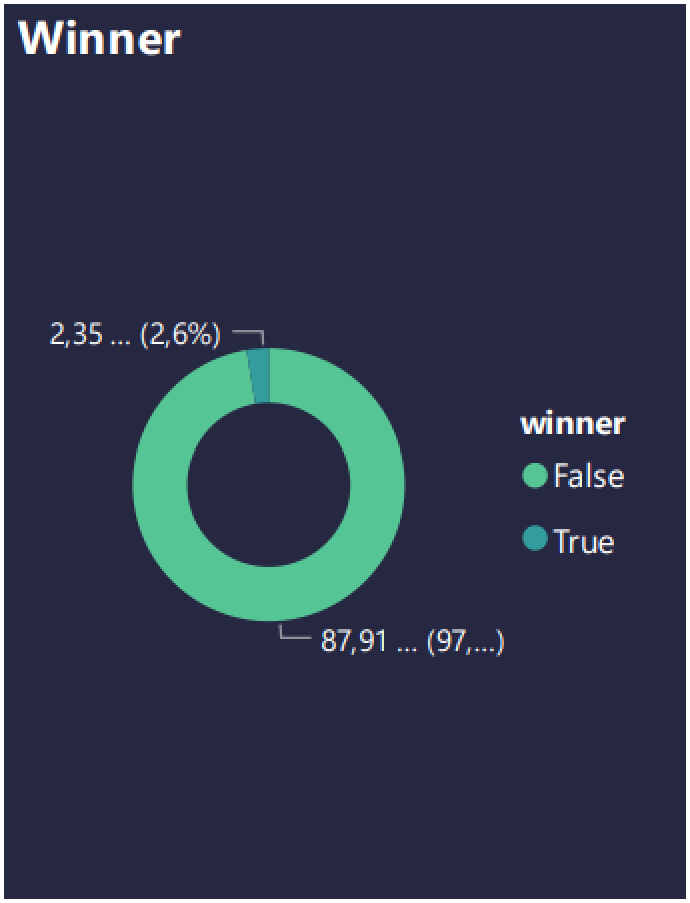


The most popular songs are:

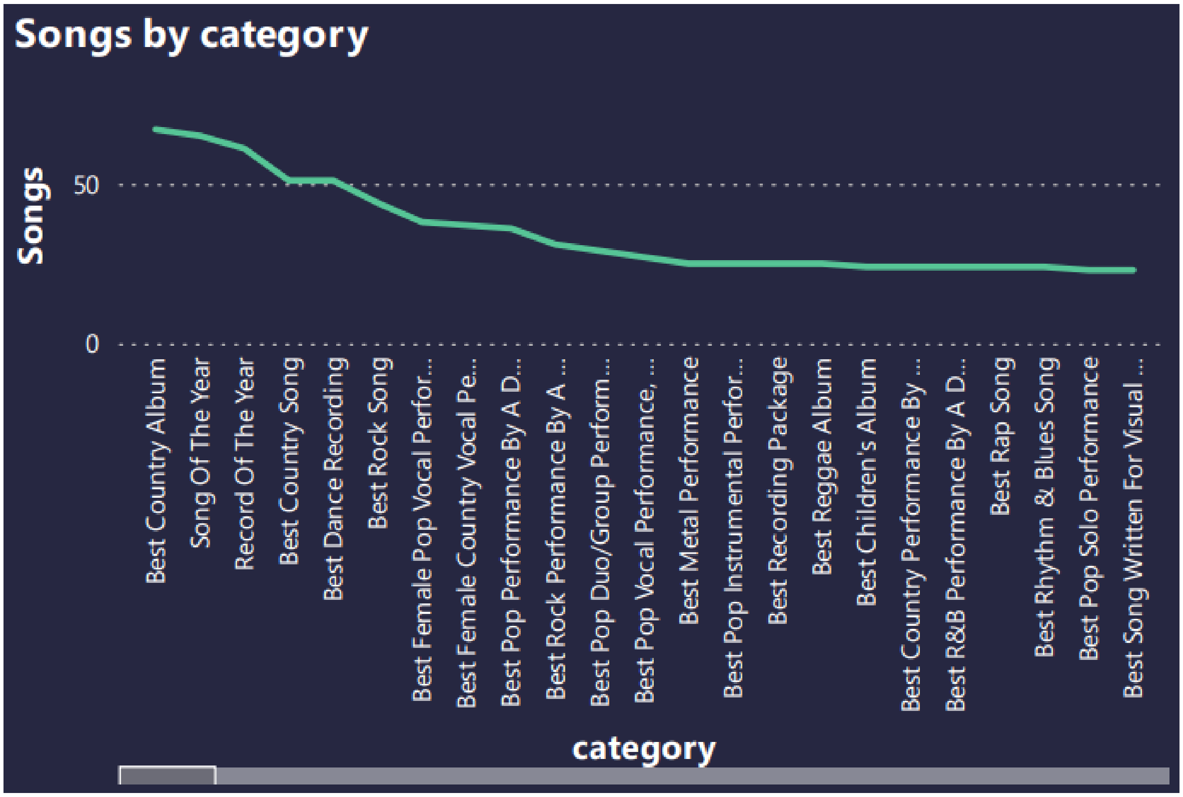
1- Unholy (feat. Kim Petras)
2- Quevedo: Bzrp Music Sessions. Vol. 52
3- La Bachata



The most quantity of artist are in the genre “Electronic/Dance”



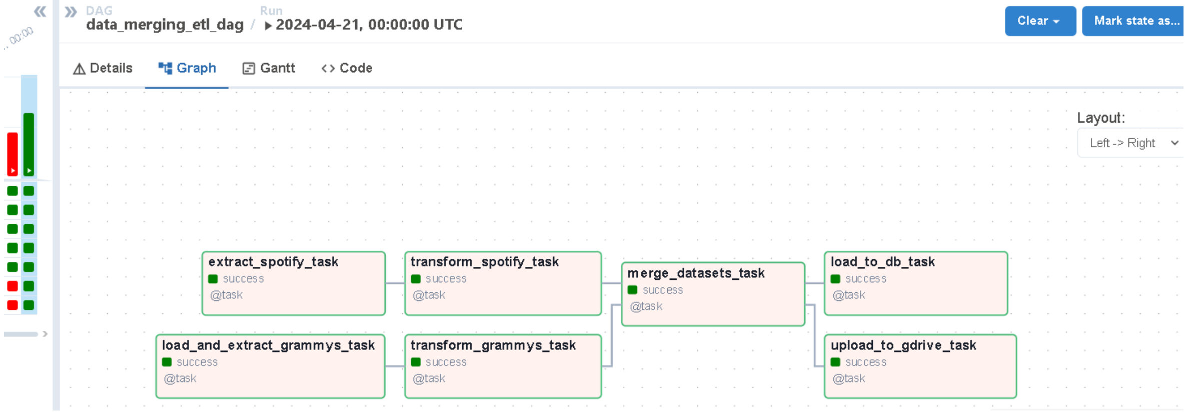
With this graph we evidence that there are less winners than “no winners” in the merge of both datasets.



Here we can look at the category with most songs in the time is “Best Country Album” And in second place we have “Song Of The Year”

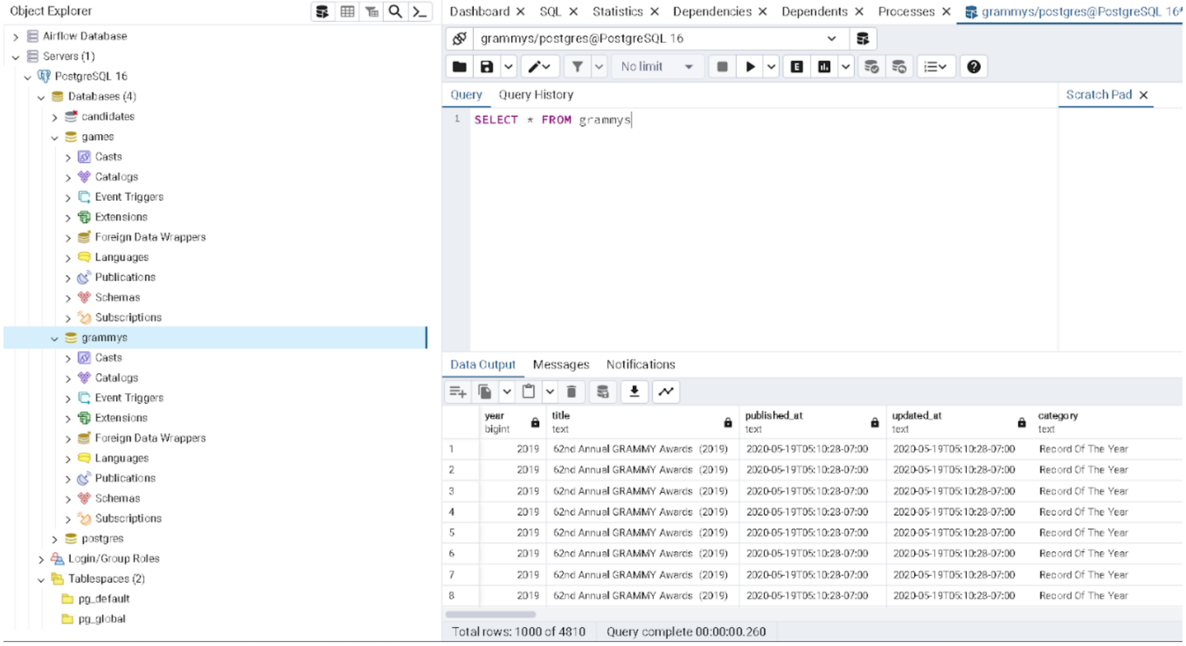
Evidence:

Worflow working:


DAG data_merging_etl_dag / Run 2024-04-21, 00:00:00 UTC

Layout: Left -> Right

Insertion of Grammys data in postgresql

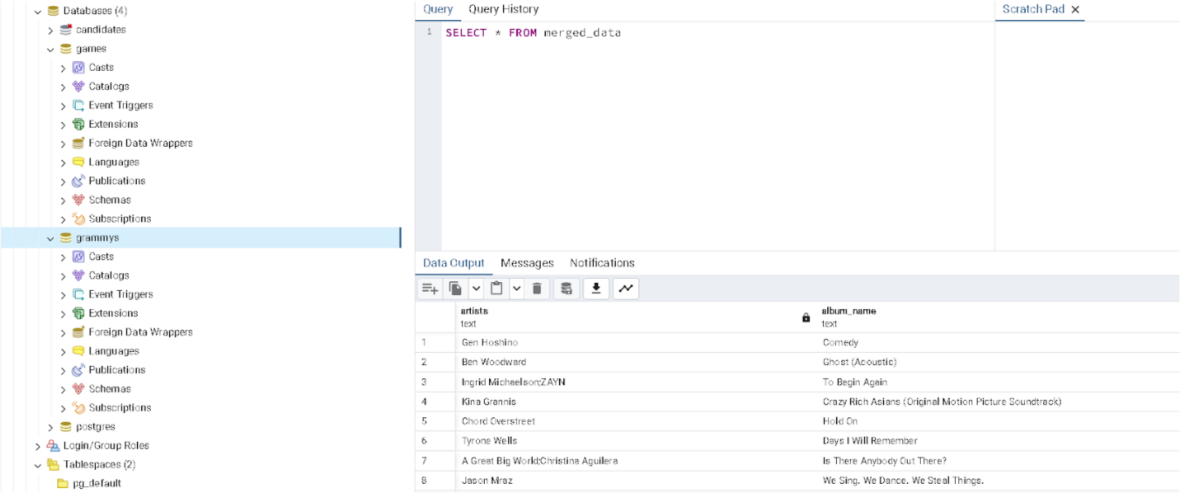


```sql
SELECT * FROM grammys
```

	year bigint	title text	published_at text	updated_at text	category text
1	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
2	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
3	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
4	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
5	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
6	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
7	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year
8	2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	2020-05-19T05:10:28-07:00	Record Of The Year

Total rows: 1000 of 4810 Query complete 00:00:00.260

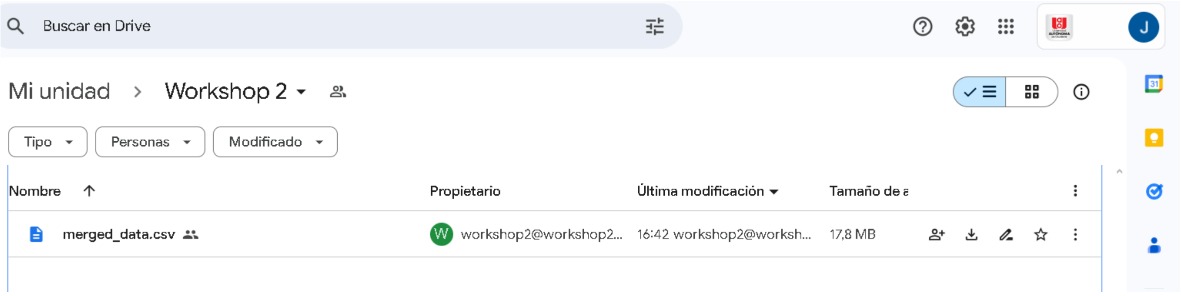
Merged data in postgresql



```sql
SELECT * FROM merged_data
```

	artists text	album_name text
1	Gen Hoshino	Comedy
2	Ben Woodward	Ghost (Acoustic)
3	Ingrid Michaelson;ZAYN	To Begin Again
4	Kina Grannis	Crazy Rich Asians (Original Motion Picture Soundtrack)
5	Chord Overstreet	Hold On
6	Tyrone Wells	Days I Will Remember
7	A Great Big World;Christina Aguilera	Is There Anybody Out There?
8	Jason Mraz	We Sing. We Dance. We Steal Things.

Data in the drive after the workflow runs:



Nombre	Propietario	Última modificación	Tamaño de a
merged_data.csv	workshop2@workshop2...	16:42 workshop2@worksh...	17,8 MB

Conclusions:

- This was a hard challenge for me, but in the process of make the etl and the dags I learned a lot of how the workflow in airflow works, I learned too to work with docker, this works without any problem to me.
- The merge was the hardest challenge for me because I didn't know how get the necessary quantities of data from both datasets, but finally I did a merge with the enough data to make a good analysis.