

juan felipe zambrano-2221031

1.CONEXIÓN A LA BASE DE DATOS

para realizar la conexión a la base de datos de datos lo primero fue crear un contenedor de docker con el comando docker compose up -d desde la terminal de visual studio, después de esto se descargó una extensión de visual llamada mysql para realizar la conexión con postgres, hay que decir que para esto se realizó la creación de un entorno virtual en anaconda para poder acceder a jupyter lab.

2.migración del csv a la base de datos

esta se realizó por medio de la librería psycopg2 pues primero se hizo la configuración por así decirlo los cuales son los parámetros, como el usuario, la clave, y la base de datos a usar para realizar la conexión la cual fue creada por medio de una función llamada create connection(), después de realizada esta conexión y verificar que este funcionado se siguió con la creación de la tabla

creacion =

```
CREATE TABLE grammy (  
    year INTEGER,  
    title VARCHAR(255),  
    published_at TIMESTAMP,  
    updated_at TIMESTAMP,  
    category VARCHAR(1000),  
    nominee VARCHAR(1000),  
    artist VARCHAR(1000),  
    workers VARCHAR(1000),  
    img VARCHAR(1000),  
    winner BOOLEAN  
);
```

3.inserción de los datos dentro de la tabla

después de creada la tabla que usaremos para hacer las gráficas insertamos los datos del csv de los candidatos dentro de esta misma iterando sobre cada una de las columnas del csv

4.leer la tabla

después de realizado dicho proceso se realizó leer la información desde la base de datos

la cual se hizo con un select * from de la tabla ya creada anteriormente la cual contenía toda la información de los grammys

5.transformaciones

En este proceso se realizaron las transformaciones de los 2 dataset para después poder hacer el merge.

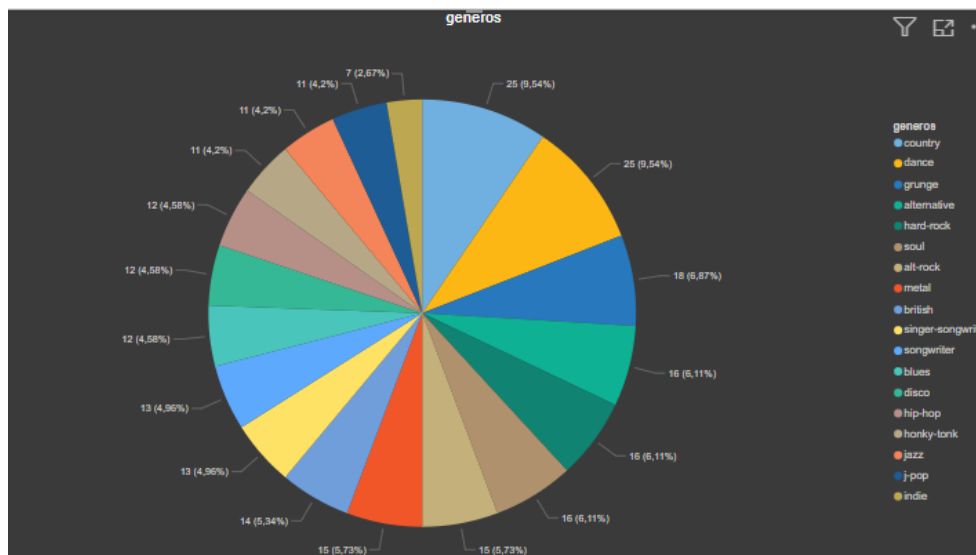
las transformaciones que se le hizo al de lo grammy fue eliminar las filas que contengan valores nulos, eliminar en la columna de artistas los nombres que estuvieran después de la coma o del punto y coma para así de esta manera dejar solo el primer nombre y también se les eliminó las siguientes columnas: title, published_at, updated_at, img, workers

en lo que respecta a spotify se realizó el mismo proceso con la diferencia de que en este se realizó la transformación de la columna de milisegundos a segundos, y pues se eliminó las siguientes columnas: Unnamed: 0, explicit, danceability, energy, duration_ms, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature, track_id, key

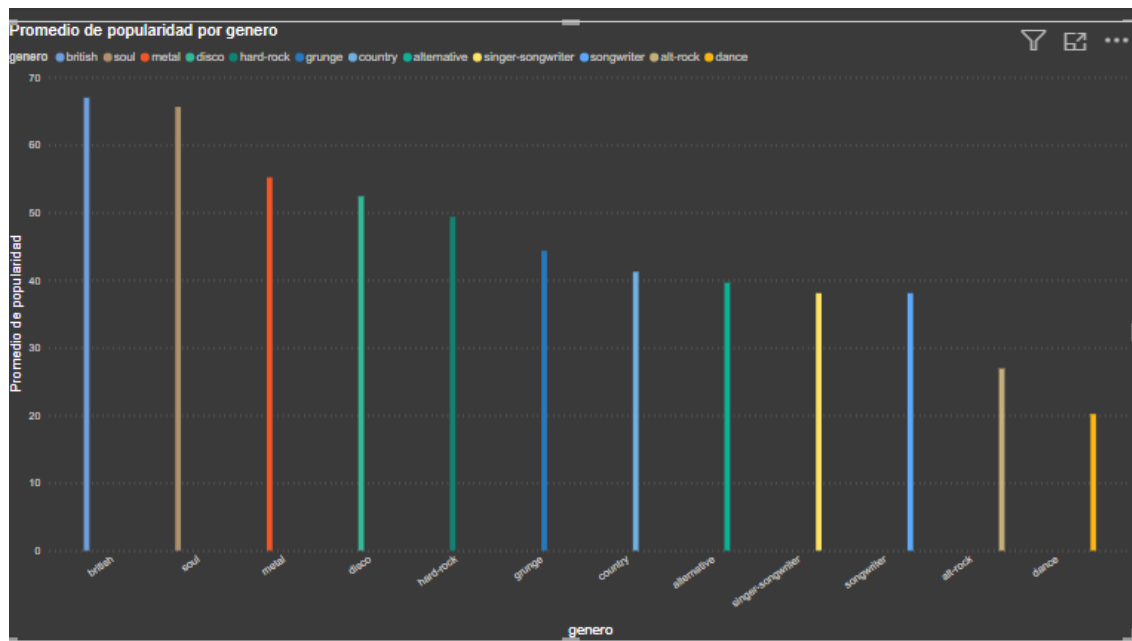
5.MERGE

Se utiliza el método merge de pandas para combinar dos DataFrames, transform grammy y transform csv, utilizando un join y un inner join. El join se realiza en dos columnas específicas: 'nominee' y 'artist' en el Data Frame transform grammy, y 'track name' y 'artists' en el Data Frame transform csv. Esto significa que solo se mantendrán las filas que tienen coincidencias en ambas columnas especificadas.

GRÁFICAS :

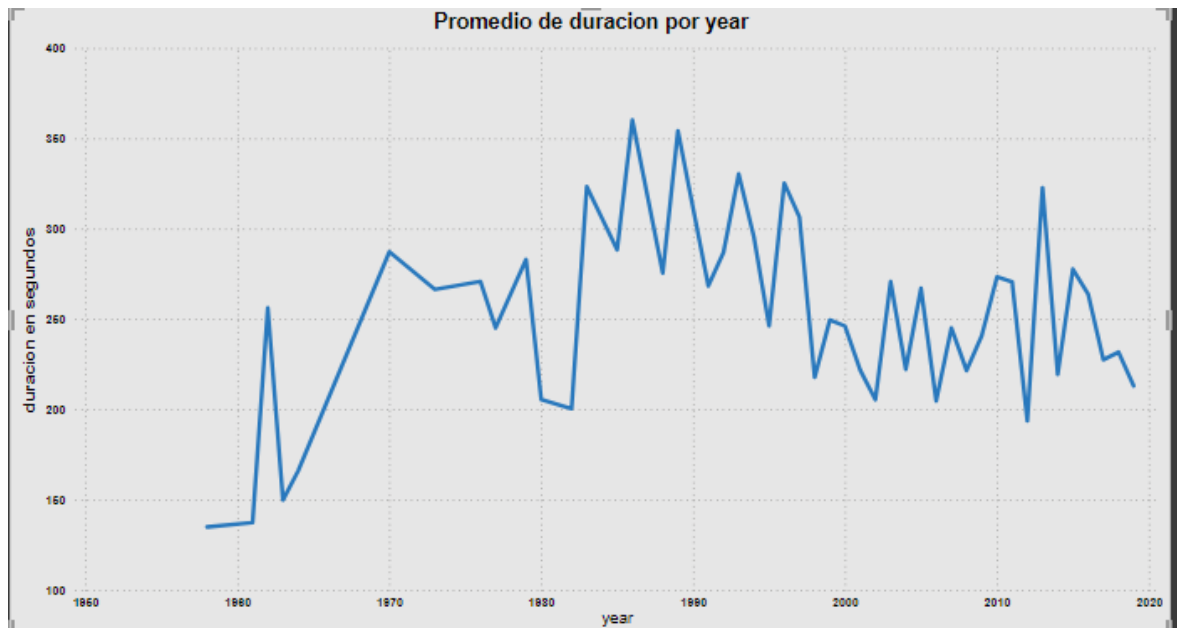


-en esta gráfica de torta son los géneros que más veces han ganado un grammy a partir de esto podemos afirmar que los géneros que más han ganado estos premios son country y dance



-en esta gráfica de barras la cual es acerca del promedio de popularidad de los géneros ganadores de los grammy aquí podemos apreciar que el de mayor promedio es el género british pero esto como lo pudimos ver con la gráfica de torta no significa que es el más ganador

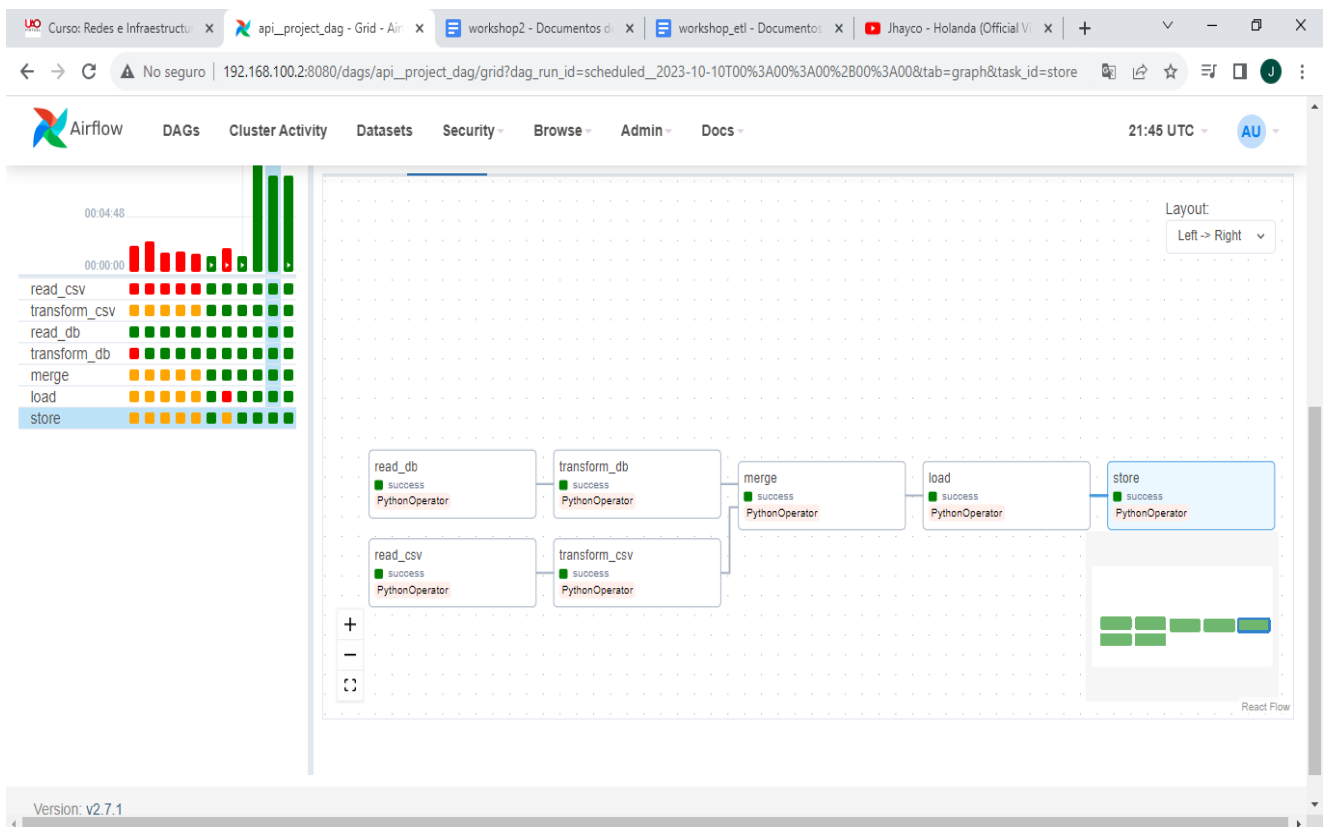
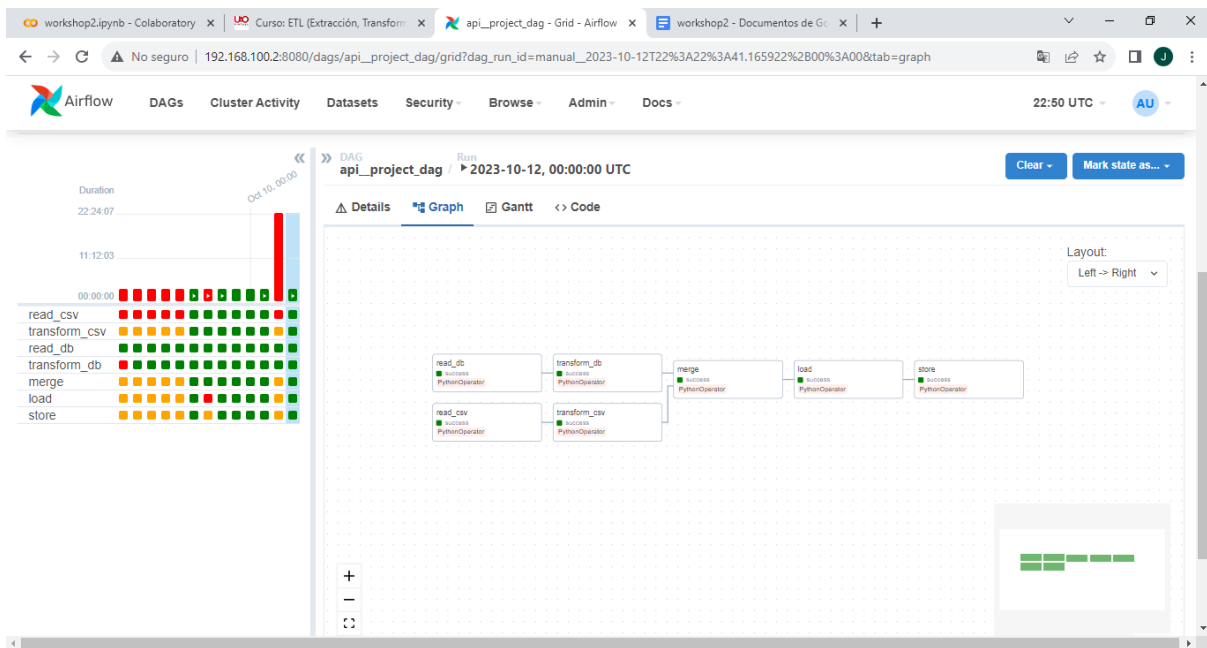




-estas 2 gráficas van relacionadas ya que en él la primera se hace promedio d la duración de las canciones ganadoras de los grammy y en la segunda es a lo largo de los años, podemos apreciar que la duración de las canciones han venido aumentando desde el inicio de los premios grammy teniendo su pico en duración entre los años 80 y 90

EVIDENCIAS

AIRFLOW :



ARCHIVOS SUBIDOS A MAQUINA VIRTUAL:

```
root@servidorUbuntu: ~/airflow_etl/api_dag
IPv4 address for br-81be95710e6e: 172.21.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for docker_gwbridge: 172.20.0.1
IPv4 address for eth0: 10.0.2.15
IPv4 address for eth1: 192.168.100.2

* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
Last login: Wed Oct 11 20:05:39 2023 from 10.0.2.2
vagrant@servidorUbuntu:~$ sudo -i
root@servidorUbuntu:~# ls
airflow db.lossalamanbiches.net ftp_server mi_web test_docker
airflow_etl dockerComposeTest gym named.conf.default-zones ubuntu_docker
appCapas docker-flask-example haproxy-docker pagina_error ubuntu_docker2
appMicro etl minecraft_data snap
root@servidorUbuntu:~# cd airflow_etl
root@servidorUbuntu:~/airflow_etl# ls
airflow.cfg airflow-webserver.pid api_dag data.csv logs webserver_config.py
airflow.db ambiente api_dag.py etl.py standalone_admin_password.txt workshop_2
root@servidorUbuntu:~/airflow_etl# cd api_dag
root@servidorUbuntu:~/airflow_etl/api_dag# ls
base_etl base_etl.py csv_etl.py merge merge.py __pycache__ spotify_dataset.csv workshop_dag.py
root@servidorUbuntu:~/airflow_etl/api_dag#
```

```
root@servidorUbuntu: ~/airflow_etl/api_dag
from datetime import timedelta
from airflow import DAG
from airflow.operators.python import PythonOperator
from airflow.models.baseoperator import chain
from datetime import datetime
from base_etl import traer_info, transformacion_granmy
from csv_etl import leer_csv, transformacion
from merge import merge, load
default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2023, 0, 13), # Update the start date to today or an appropriate date
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=1)
}

def func1():
    print(f'the date is: {datetime.now()}')

with DAG(
    'api_project_dag',
    default_args=default_args,
    description='Our first DAG with ETL process!',
    schedule_interval='@daily', # Set the schedule interval as per your requirements
) as dag:

    merge = PythonOperator(
        task_id='merge',
        python_callable=merge,
        provide_context = True,
    )

    read_csv = PythonOperator(
        task_id='read_csv',
        python_callable=leer_csv,
        provide_context = True,
    )

    transform_csv = PythonOperator(
        task_id='transform_csv',

```

```
root@servidorUbuntu: ~/airflow_etl/api_dag
import pandas as pd
import json
import logging
import re

def leer_csv():
    df = pd.read_csv("./api_dag/spotify_dataset.csv")
    df.dropna(inplace=True)
    return df.to_json(orient='records')

def transformacion(**kwargs):
    ti = kwargs["ti"]
    str_data = ti.xcom_pull(task_ids='read_csv')
    json_data = json.loads(str_data)
    df = pd.json_normalize(data=json_data)
    # Función para eliminar coma
    def eliminar_coma(entrada):
        return re.sub(r',', '.', entrada)

    # Función para eliminar punto
    def eliminar_punto(entrada):
        return re.sub(r';', '.', entrada)

    # Eliminar coma y eliminar punto para conservar solo el primer artista
    df['artists'] = df['artists'].apply(eliminar_coma)
    df['artists'] = df['artists'].apply(eliminar_punto)

    # Calcular la duración en segundos
    df['duration_seconds'] = df['duration_ms'] / 1000

    # Eliminar columnas no deseadas
    columnas_a_eliminar = ['Unnamed: 0', 'explicit', 'danceability', 'energy', 'duration_ms', 'loudness',
                           'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence',
                           'tempo', 'time_signature', 'track_id', 'key']
    df = df.drop(columnas_a_eliminar, axis=1)

    return df.to_json(orient='records')

"csv_etl.py" [dos] 37L, 1328B 1,1 All

root@servidorUbuntu: ~/airflow_etl/api_dag
import psycopg2
import pandas as pd
import re
import json

def traer_info():
    # Establecer la conexión a la base de datos y obtener los datos
    conn = psycopg2.connect(
        host='192.168.188.1',
        user='postgres',
        password='admin',
        database='etl_db'
    )
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM grammy')
    rows = cursor.fetchall()
    columns = [col[0] for col in cursor.description]
    df = pd.DataFrame(rows, columns=columns)
    conn.close()

    return df.to_json(orient='records')

def transformacion_grammy(**kwargs):
    ti = kwargs["ti"]
    str_data = ti.xcom_pull(task_ids='read_db')
    json_data = json.loads(str_data)
    df = pd.json_normalize(data=json_data)
    # Eliminar Filas con valores nulos en la columna 'artist'
    df = df.dropna(subset=['artist'], inplace=True)

    # Función para eliminar coma
    def eliminar_coma(entrada):
        return re.sub(r',', '.', entrada)

    # Función para eliminar punto
    def eliminar_punto(entrada):
        return re.sub(r';', '.', entrada)

    # Aplicar eliminar_coma y eliminar_punto para el campo 'artist'
    df['artist'] = df['artist'].apply(eliminar_coma)
    df['artist'] = df['artist'].apply(eliminar_punto)

    # Eliminar columnas no deseadas

"base_etl.py" [dos] 49L, 1508B 1,1 Top
```

TABLAS SUBIDAS:

File Edit Selection View Go Run ... workshop_2

merge.py 9+ csv_etl.py 1 grammy x

SELECT * FROM grammy LIMIT 100

Search results Cost: 1s24ms 1 2 3 4 ... 97 Total 9620

	year integer	title character varying(255)	published_at timestamp without time z	updated_at timestamp without time z	category character varying(1000)	nominee character varying(1000)	artist character varying(1000)
1	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Bad Guy	Billie Eilish
2	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Hey, Ma	Bon Iver
3	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	7 rings	Ariana Grande
4	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Hard Place	H.E.R.
5	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Talk	Khalid
6	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Old Town Road	Lil Nas X
7	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Truth Hurts	Lizzo
8	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Record Of The Year	Sunflower	Post Malone
9	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	When We All Fall Asleep, Where Do We Go?	Billie Eilish
10	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	LI	Bon Iver
11	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	Norman F***ing Rockwell	Lana Del Rey
12	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	thank u, next	Ariana Grande
13	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	I Used To Know Her	H.E.R.
14	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	7	Lil Nas X
15	2019	62nd Annual GRAMMY Awards	2020-05-19 05:10:28	2020-05-19 05:10:28	Album Of The Year	Cuz I Love You (Deluxe)	Lizzo

9 4 0 postgres etl_db

File Edit Selection View Go Run ... workshop_2

merge.py 9+ csv_etl.py 1 merge x

SELECT * FROM merge LIMIT 100

Search results Cost: 401ms 1 2 3 4 Total 363

	year integer	category character varying(1000)	nominee character varying(1000)	artist character varying(1000)	winner boolean	album_name character varying(1000)	track_name character varying(1000)	popularity integer	track_genre character varying(1000)
1	2019	Record Of The Year	7 rings	Ariana Grande	true	thank u, next	7 rings	84	dance
2	2019	Record Of The Year	7 rings	Ariana Grande	true	thank u, next	7 rings	84	pop
3	2019	Best Pop Solo Performance	7 rings	Ariana Grande	true	thank u, next	7 rings	84	dance
4	2019	Best Pop Solo Performance	7 rings	Ariana Grande	true	thank u, next	7 rings	84	pop
5	2019	Record Of The Year	Truth Hurts	Lizzo	true	Give You Love - Cozy Hits	Truth Hurts	1	hip-hop
6	2019	Record Of The Year	Truth Hurts	Lizzo	true	Rap Party	Truth Hurts	0	hip-hop
7	2019	Record Of The Year	Truth Hurts	Lizzo	true	Good Enough - Easy Pop	Truth Hurts	0	hip-hop
8	2019	Best Pop Solo Performance	Truth Hurts	Lizzo	true	Give You Love - Cozy Hits	Truth Hurts	1	hip-hop
9	2019	Best Pop Solo Performance	Truth Hurts	Lizzo	true	Rap Party	Truth Hurts	0	hip-hop
10	2019	Best Pop Solo Performance	Truth Hurts	Lizzo	true	Good Enough - Easy Pop	Truth Hurts	0	hip-hop
11	2019	Album Of The Year	thank u, next	Ariana Grande	true	thank u, next	thank u, next	82	dance
12	2019	Best Pop Vocal Album	thank u, next	Ariana Grande	true	thank u, next	thank u, next	82	dance
13	2019	Best Pop Duo/Group Performance	Sucker	Jonas Brothers	true	Happiness Begins	Sucker	81	dance
14	2019	Best Pop Vocal Album	Lover	Taylor Swift	true	Autumn Vibes 2022	Lover	0	pop
15	2019	Best Pop Vocal Album	Lover	Taylor Swift	true	Lover	Lover	85	pop

9 4 0 postgres etl_db

conexion base de datos power bi:

ArchivoInicioInsertar

Obtener datos

neros
country
dance
grunge
alternative
hard-rock
soul
alt-rock
metal
british
singer-songwriter
songwriter
blues
disco
hip-hop
honky-tonk
jazz

Promedio de popularidad

genero brit

60
40
20
0

british

Página 3

Seleccionar tablas relacionadas

Navegador

localhost: etl_db [2]

public.grammy

public.merge

No hay elementos seleccionados para vista previa

Cargar

Transformar datos

Cancelar

Inicio sesión

Publicar

Datos

Buscar

public merge

datos a...

de ob...

Página 1 de 2

100 %