Documento ETL Workshop 2

Isaac Piedrahita (2226259)

Cuarto Semestre de Ingeniería de Datos e Inteligencia Artificial

Javier Alejandro Vergara Zorrilla

ETL (Extract, transform and load)

Ingeniería de datos e inteligencia artificial, Facultad de ingeniería

Universidad Autónoma de Occidente

Santiago de Cali

2024

Documento de Evidencia - Workshop 2 - ETL

Este documento detalla el proceso y las técnicas utilizadas en el proyecto ETL_Workshop2_Challenge, que fue diseñado para construir una pipeline de ETL utilizando datos de múltiples fuentes. El objetivo principal era extraer datos de fuentes variadas, transformarlos adecuadamente y cargarlos en un almacenamiento final como Google Drive y una base de datos, para luego realizar visualizaciones efectivas sobre los datos.

Descripción del Repositorio
El repositorio consta de varios directorios y archivos, centrados principalmente en scripts de Python que facilitan las operaciones de ETL:

Tecnologías utilizadas
Python:
Todos los scripts de transformación y carga están escritos en Python.
PostgreSQL:
Para el desarrollo del proyecto se escogió PostgreSQL Serverless alojada en Neon como solución de base de datos debido a sus capacidades de escalabilidad automática y eficiencia en la gestión de costos. Este entorno serverless facilita una mayor concentración en las tareas analíticas sin la necesidad de administrar la infraestructura subyacente, lo que resulta ideal para manejar los volúmenes de datos fluctuantes típicos de los procesos ETL.
Looker Studio:
Looker Studio se implementó para crear dashboards interactivos y permitió transformar los datos procesados en representaciones visuales comprensibles. Las visualizaciones desarrolladas ayudan a destacar tendencias significativas, correlaciones y patrones entre los datasets de Spotify y Grammy.
PyDrive:
PyDrive se utilizó para automatizar la carga de los archivos CSV resultantes del proceso de merge directamente en Google Drive.

Directorios
- code/: Contiene scripts de Python que implementan las transformaciones de datos y la lógica de ETL.
- **data**/, **raw_data**/: Destinados a contener los datasets necesarios y los datos en bruto.
- **transformations**/: Alberga los scripts específicos que transforman los datasets antes de su carga final.

Análisis de Scripts de Transformaciones:

Grammy_Transformations.py

Este script maneja las transformaciones de los datos del dataset de los Grammy Awards. El proceso incluye la limpieza de datos, la normalización de campos y la preparación de estos para su integración con otros datasets. Ejemplo:

```
import pandas as pd

def clean_grammy_data(df):
    # Limpiar nombres y categorías
    df['name'] = df['name'].str.title()
    df['category'] = df['category'].str.upper()
    return df

# Código para cargar y transformar datos
df = pd.read_csv('grammy_data.csv')
df_cleaned = clean_grammy_data(df)
df_cleaned.to_csv('grammy_cleaned.csv', index=False)
```

Spotify_Transformations.py

Este archivo se centra en los datos de Spotify, aplicando transformaciones similares para garantizar que los datos estén listos para ser combinados con otros sets. Ejemplo:

```
def transform_spotify_tracks(df):
    # Convertir milisegundos a minutos
    df['duration'] = df['duration_ms'] / 60000
    return df.drop('duration_ms', axis=1)

df_spotify = pd.read_csv('spotify_tracks.csv')
df_transformed = transform_spotify_tracks(df_spotify)
df_transformed.to_csv('spotify_transformed.csv', index=False)
```

Merge.py

Encargado de combinar datos de diferentes fuentes en un único dataset coherente, este script es crucial para preparar la carga final:

```
def merge_datasets(df1, df2):
    # Unir datasets en una sola tabla
    return pd.merge(df1, df2, on='id', how='inner')

df_grammy = pd.read_csv('grammy_cleaned.csv')
df_spotify = pd.read_csv('spotify_transformed.csv')
df_merged = merge_datasets(df_grammy, df_spotify)
df_merged.to_csv('final_dataset.csv', index=False)
```

EDAs: Algunas Conclusiones

Spotify dataset:

Distribución de la Popularidad de las Pistas:

- Las puntuaciones de popularidad están sesgadas hacia la derecha, lo que sugiere que la mayoría de las pistas tienen puntuaciones bajas.
- Hay un pico significativo en la puntuación de 0, indicando que muchos temas no son populares.
- La cantidad de pistas disminuye gradualmente a medida que aumenta la popularidad, siendo muy pocas las que alcanzan una puntuación cercana a 100.

Distribución de la Duración de las Pistas:

- La duración de las pistas muestra valores atípicos, ya que hay puntos dispersos hacia el extremo derecho del eje x.
- La mayoría de las pistas se agrupan dentro de un rango de 200,000 a 300,000 milisegundos (200 a 300 segundos), típico para la duración estándar de las canciones.

Proporción de Contenido Explícito:

- Un pequeño porcentaje (aproximadamente 8.6%) de las pistas están marcadas como explícitas.
- La gran mayoría de las pistas (91.4%) no son explícitas, lo que sugiere una tendencia de contenido apto para toda la familia.

Distribución de la Bailabilidad:

- La característica de bailabilidad sigue una distribución normal, indicando que la mayoría de las pistas tienen un nivel moderado de bailabilidad.
- El pico de la distribución se encuentra alrededor de 0.6 a 0.7, lo que significa que muchas pistas son bastante bailables.

Distribución de la Energía:

- La característica de energía también sigue una distribución normal, pero con una ligera inclinación hacia niveles de energía más altos.
- La mayoría de las pistas tienen una calificación de energía de alrededor de 0.6 a 0.8, lo que sugiere la presencia de pistas más energéticas en el dataset.

Distribución de la Valencia:

- La característica de valencia muestra una distribución multimodal, lo que indica la presencia de grupos de pistas con diferentes niveles de positividad musical.
- La distribución tiene picos en el extremo inferior y hacia el medio, con menos pistas que tienen una valencia muy alta.

Distribución de los Principales Géneros:

- La distribución de géneros muestra una representación diversa.
- El gráfico circular sugiere una distribución bastante equitativa entre los géneros principales, cada uno ocupando alrededor del 10% del dataset, aunque sin números exactos, esto es una estimación.
- La diversidad real en género podría ser mayor o menor de lo sugerido debido al número limitado de géneros mostrados en el gráfico circular.

Grammy Dataset

Categorías de Premios Más Galardonadas:

- Las categorías de "Canción del Año", "Grabación del Año" y "Álbum del Año" tienen la mayor cantidad de premios, lo que refleja su prestigio y la alta competencia en estas áreas.
- Categorías como "Mejor Álbum de Música de Cámara" y "Mejor Álbum Histórico" tienen menos premios, lo que podría indicar nichos más especializados o una menor cantidad de entradas en estas categorías.

Distribución de Premios por Año:

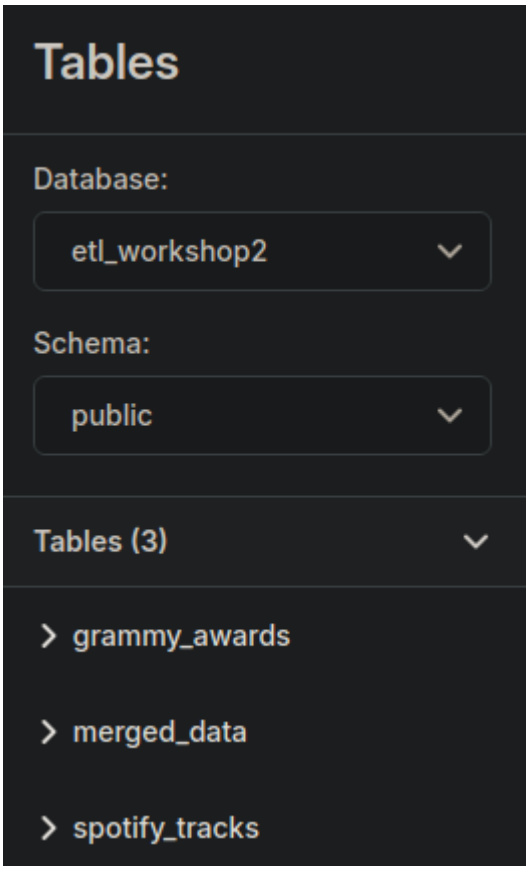
- Hay un aumento dramático en el número de premios otorgados en 2019, lo que podría ser resultado de un cambio en las reglas del Grammy, la adición de nuevas categorías, o un evento excepcional ese año.
- A lo largo de los años, ha habido un aumento gradual en la cantidad de premios otorgados, lo que podría reflejar la expansión de la industria musical y el reconocimiento de más géneros y artistas.

Top de Nominados y Ganadores:

- Obras como "Berlioz: Requiem" y "Bridge Over Troubled Water" figuran entre las más nominadas, demostrando su impacto duradero en la música.
- El término "Desconocido" lidera en la categoría de los artistas más premiados con 1840, lo cual es probablemente un marcador de lugar para agrupar los ganadores no identificados o diversos artistas en categorías grupales.
- Entre los artistas reconocibles, U2 y Aretha Franklin están entre los más premiados, lo que muestra su éxito y la calidad de su música constante a lo largo del tiempo.

Evidencias:

Tablas alojadas en base de Datos PostgreSQL




```
1 SELECT * FROM grammy_awards;
```

Ready (1 query)

Run | Explain | Analyze | 757ms | 4810 rows

#	year	title	published_at	updated_at	category
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

```
1 SELECT * FROM spotify_tracks;
```

Ready (1 query)

Run | Explain | Analyze — 1s — 114000 rows

#	track_id	artists	album_name	track_nan
1	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy
2	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic
3	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again
4	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original	Can't Hel Falling In

```
1 SELECT * FROM merged_data;
```

Ready (1 query)

Run | Explain | Analyze — 1s — 24462 rows

#	year	title	published_at	updated_at	category	nor
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	Bac

CSV alojado en Google Drive (usando mismo ID de carpeta que en el código con que fue cargado el archivo)

```
1  SELECT * FROM merged_data;
```

Ready (1 query)

Run | Explain | Analyze | 1s | 24462 rows

#	year	title	published_at	updated_at	category	nor
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	Bac

mail.google.com/mail/u/1/#search/jalvergara%40uao.edu.co/FMfcgzGxSHkDgkClPVdfWWgMkdSPkJnL

```python
from pydrive2.auth import GoogleAuth
from pydrive2.drive import GoogleDrive

def authenticate():
    gauth = GoogleAuth()
    gauth.LoadClientConfigFile("pyDrive/client_secrets.json")
    gauth.LocalWebserverAuth()
    return gauth

def upload_to_drive(file_path, folder_id='1zaubLuFwaNfrGQHC5B8Zpn1SPi3IMpXB'):
    gauth = authenticate()
    drive = GoogleDrive(gauth)

    file_name = file_path.split('/')[-1]
    file = drive.CreateFile({'title': file_name, 'parents': [{'id': folder_id}] if folder_id else []})
    file.SetContentFile(file_path)
    file.Upload()
    print(f"Archivo '{file_name}' subido correctamente a Google Drive.")

file_path = "data/merged_data.csv"
upload_to_drive(file_path)      You, 1 hour ago • merge-to-drive&dashboard
```