ETL Group 4

Project Scope:

Github Repository : ETL Project Group 4

Where is the data coming from? (at least two sources)

- Apple Music API music playlists (https://api.music.apple.com/v1/me/library/playlists/{id})
- Spotify API music playlists (client_id = "a753d56129f44f3fb9f17e0061f38eba) curl -H "Authorization: Basic a753d56129f44f3fb9f17e0061f38eba:70e3adb4ad4e45d4a27c65a576d2d411" -d grant_type=authorization_code -d code=AQDscVS9sv5-DBQ-Zh3fdXY7YM8zkaIDYEnxoNrg6OG_cp2FoV8B4qpG66YfNIrcqWvmD3ZvjhI38I6bDAs-smE_UWN7WSL3QRiKFDkOKgsxT5O-LbE-TQsM3hWKxMSxliJ5hxXVYgixq3VgA_Z_3_Pa0kNpq_ImIBkeRbd1f-78UflW6XnXYA -d redirect_uri=https%3A%2F%2Fgoogle.com https://accounts.spotify.com/api/token

- Grammy’s (https://www.kaggle.com/unanimad/grammy-awards)

Where is the data going to? (postgres, mongo, etc; some type of database not a flat file like a CSV.)
postgres

What will be the structure of the data in the final database? What tables/columns/types/etc.

End goal to display multiple top 5’ playlists denoted by characteristic,

Song title	String
Artist name	String
User streams	INT
Release Date	FLOAT
Added Date	FLOAT
Song of the Year	String
Grammy winner?	Boolean
Number of playlists	INT

Random Thoughts:

Add tracks to a new playlist on the user's Spotify account

Are there genres or characteristics that shine on one platform over the other.
2006 – spotify
2003 – iTunes/Apple Music - 2015

***Final product creates new Spotify playlist

How do we account for purity of data? – limit to specific playlist

Anna

1. View authorization steps for Spotify API authorization
    a. Discovered OAuth examples Github repository and reviewed before installing Node.js
    b. Installed node.js
    c. Obtained

Kara

1. Downloaded Past Grammy Winners CSV from Kaggle
2. Imported to Pandas and saved only the Grammy categories for individual songs into a data frame.
3. Removed years not relevant to the Spotify data
4. Renamed 'nominee' column header to 'title'
5. Exported to a new CSV and pushed to a git branch

Lisa

1. View authorization steps for Apple Music API
    a. Create apple certificate
    b. Create private key
    c. Manage identifiers
    d. Troubleshoot

2. Created branch for git work
3. Started py doc for import.
4. Researched other finished projects for help.
    a. https://github.com/rcrdclub/apple-py-music.git
    b. https://github.com/mpalazzolo/apple-music-python.git
    c.
5. Apple Music Developer
    a. Create membership
    b. Get Team_ID
    c. Create certificate
    d. Create identifier
    e. Create api key
6. Learning and installing modules via terminal
    a. Applemusicpy
    b. apple-music-python
    c. jwt
    d. Pylint
    e. Requests
    f. Updated pip
7. Installing manually
    a. Pylint
8. Restart from scratch using class resources for guidance.

Project Revision:

As both the Apple Music and Youtube API's presented a challenge, we are re-focusing our research.

Databases:
https://www.kaggle.com/unanimad/grammy-awards
https://www.kaggle.com/leonardopena/top-spotify-songs-from-20102019-by-year

Goal:
Organize data from multiple databases in order to determine similar characteristics of Grammy-winning songs between the years of 2010 and 2019 per streaming platform Spotify.

Steps:
Get CSV into postgres
    -Latin1
Grammy Table
    Columns:
    Year (year),
    Category(category),
    Song Title (title),
    Workers(workers),
    Winner (winner)
Spotify Table
    Columns:
    ID (id)
    Year (year)
    Genre (top_genre)
    Artist (artist)
    Song’s Title (title)
    Beats Per Minute (bpm)
    Energy (nrgy)
    Danceability (dnce)
    Loudness (db)
    Liveness (live)
    Valence (val)
    Duration in seconds (dur)
    Accousticness (acous)
    Speachiness (spch)
    Popularity (pop)

Inner Join on Song Title