**TOP MUSIC OF 2023: A SQLITE DATABASE**

This project is focused on the development of a web application, served on Flask, that integrates with Python code for querying. The focus extends beyond functionality to encompass a visually appealing and user-friendly experience, achieved through the implementation of HTML and CSS for development and styling. Additionally, SQL was utilized to create an SQLite database that forms the backbone of the application.

The web app is designed to facilitate easy and efficient data retrieval, allowing users to use query results that interact with the Flask backend. Using the power of Flask, Python, HTML, and CSS, this project creates a harmonious blend of functionality and aesthetics, providing users with an interactive platform that is not only robust in its querying capabilities but also visually polished and engaging.

**Data Source:**

The dataset for this analysis was obtained from Kaggle - <https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023>.

The dataset contains 953 records with fields track name, artist(s) name, number of artists, year released, month released, day released, total number of streams on spotify, number of spotify playlists the song is included in, presence and rank of the song on spotify charts, number of Apple Music playlists the song is included in, presence and rank of the song on Apple Music charts, number of Deezer playlists the song is included in, presence and rank of the song on Deezer charts, presence and rank of the song on Shazam charts, beats per minute, key of the song, mode of the song (major or minor), danceability, valence, energy, acousticness, instrumentalness, liveness, and speechiness.

**Data Cleaning and Preparation:**

\* *Handling missing values*

o Other than columns in\_shazam\_charts and key\_of\_song, missing some values, all the records in the dataset have all the values.

\* *Duplicate values*

o It is possible that more than one artist used same track name for their songs. Therefore, songs with same track name but different artists were not considered duplicate in this analysis. However, songs with same track name and same artist(s) were considered duplicate and were removed.

\* Some of the column names were renamed for consistency.

**Using and interacting with the project:**

**Jupyter Notebook**:

Read the dataset using a Jupyter Notebook file.

Utilize SQL commands to create tables and define the schema.

**SQLite Database Creation:**

Use the SQL schema to create an SQLite database.

**Flask App Setup:**

Set up a Flask app with the necessary dependencies.

Define functions to connect to the SQLite database.

**HTML Templates and CSS:**

Create HTML templates for rendering the web pages.

Create CSS templates for styling.

**Flask Routes and Views:**

Define Flask routes to handle different URLs.

Create views to fetch data from the SQLite database.

**Run the Flask App:**

Execute the Flask app using the python spotifyApp.py command.

Web Browser Interaction:

Open a web browser and navigate to http://127.0.0.1:5000/ to view the Flask app.

For interaction with the project, users can clone the repository to the terminal and run the python file to open the flask app in the browser. To edit the query results page with your own query requests, open the python file using your preferred text editor (ex: visual studio code) and edit the "names()" function line for "results" with your specific query. Save the file and refresh the flask app page in your browser.

**Documentation of the database:**

In this project, SQL was used since it is designed for managing and querying structured data. It provided a standardized way to interact with relational databases, making it easier to retrieve, insert, update, and delete data. SQL databases support constraints and rules that help maintain data integrity. This includes primary key constraints, foreign key relationships, and unique constraints, ensuring that the data is accurate and consistent. SQL also has plenty of resources, documentation, and expertise available.

**Entity Relationship Diagram:**

**A screenshot of a computer

Description automatically generated**

**Data Analysis:**

The analysis sought to answer the following key questions:

1. Who are the top 10 artists by track count?

2. What is the distribution of tracks by key?

3. What is the listener experience by:

- Acousticness vs energy

- Acousticness vs danceability

4. Who are the top 10 artists in the charts by track counts?

5. Which are the top 10 tracks in Spotify playlists?

**Findings:**

1. From our analysis Taylor Swift had the highest number of top tracks out of any other artist in 2023. The Weeknd came next with 22 tracks followed by Bad Bunny and SZA who tied at 20 tracks each.These numbers represent the the number of times one of these artists had a song that appeared as a top spotify track.

2. The options of the keys in the top tracks were C#, G, G#, F, B, D, A, F#, E, A#, and D#. There is a somewhat linear distribution here. C# was the most popular key of top Spotify songs in 2023. C# is associated with despair, wailing, and weeping. So sad music was really popular among top Spotify tracks.

3. The analysis showed that energy and acousticness are correlational in top Spotify songs in 2023. Energy is how intense or active a song is, and acousticness relates to the songs that use instruments with minimal or no electrical interference. An example of an acoustic song is a song that uses just a guitar(no synths or ‘electrical’ sounds). Songs that are high in energy typically are not and songs that are high in acousticness tend not to be energetic. Meanwhile, the linear regression estimated that the linear relationship between Energy and Acousticness shows a negative mathematical correlation between those two values. Meaning there is a statistical probability that songs which are high in acousticness tend to have low energy. The P value in the linear regression is 8.339975569548671e-86, which is extremely small. In statistical hypothesis testing, a small p value suggests strong evidence against a null hypothesis. Since the p value is low, there is a significant difference or effect between Acousticness and Energy.

4. For Danceability and Acousticness, the linear regression proves there is a statistically significant relationship. The p value is 1.5160208603259472e-13 which is small but not very small. Since there is a somewhat small p value for this data, we can statistically prove there is a correlation between danceability and acousticness. This relationship is negative meaning the more acoustic the song is, the less danceable it tends to be.

5. In the Spotify, Apple, Shazam and Deezer charts, Taylor Swift emerged as the top artist who had the most tracks in most of the charts.

**Ethical Considerations:**

In conducting the analysis of the top streamed songs on Spotify, this project places a strong emphasis on ethical considerations. We prioritize user privacy by ensuring the dataset excludes personally identifiable information.

Transparent communication regarding data sources and methodology is communicated.

Respecting ownership and licensing agreements from Kaggle is considered, accompanied by a commitment to cultural sensitivity to avoid perpetuating stereotypes.

Security measures safeguard the dataset, and a clear definition of the analysis's purpose aligns with ethical standards.

Responsible data use is emphasized to prevent harm to individuals, communities, or artists, fostering openness about limitations and uncertainties while being accountable for addressing potential biases.

**Contributors:**

\* Aiden Tariku

o https://github.com/aidentariku

\* Naomi Baxter

o https://github.com/nbaxter6

\* Karl Shultz

o https://github.com/karl-schultz

\* Susan Miyengi

o https://github.com/smiyengi

**References:**

1. MDN Web Docs - CSS width Property: https://developer.mozilla.org/en-US/docs/Web/CSS/width

2. W3Schools - CSS text-align Property: https://www.w3schools.com/css/css\_align.asp

3. MDN Web Docs - CSS background-image Property: https://developer.mozilla.org/en-US/docs/Web/CSS/background-image

4. W3Schools - CSS3 Gradients: https://www.w3schools.com/css/css3\_gradients.asp

5. Dev.to - 3 Ways to Display Two Divs Side by Side: https://dev.to/dawnind/3-ways-to-display-two-divs-side-by-side-3d8b#:~:text=The%20most%20common%20way%20to,using%20inline%2Dblock%20css%20property.&text=The%20inline%2Dblock%20property%20on,like%20an%20inline%20element%20does.

6. W3Schools - HTML Tables: https://www.w3schools.com/html/html\_tables.asp

7. DigitalOcean - How To Use Templates in a Flask Application: https://www.digitalocean.com/community/tutorials/how-to-use-templates-in-a-flask-application

8. Color Palette - Color Hex: https://www.color-hex.com/color-palette/103731

9. Color Palette - Color Hex: https://www.color-hex.com/color-palette/70361

10. Nidula Elgiriyewithana, <https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023/data>

11. Ali Lakhani – Instructor, Data Analysis BootCamp