**Project Title: Spotify Analytics Database**

**Project Description:**

The Spotify Analytics Database project is designed to simulate a music streaming platform's database, similar to Spotify. It stores information about users, artists, songs, playlists, and listening history, enabling analysis of user behavior, song popularity, and playlist trends. The goal of this project is to provide a structured SQL database that allows data analysts to explore insights into how users interact with songs and playlists, the popularity of artists and genres, and other key metrics.

This database structure serves as a foundation for building meaningful queries that can be used to understand trends, optimize recommendations, and assess user engagement on a music platform.

**Database Structure**

The database consists of the following main tables:

1. **Users**: Stores information about each user, including username, email, country, subscription type (e.g., Free or Premium), and the date they joined the platform. This table helps in identifying user demographics and subscription types, which is essential for segmenting users for further analysis.
2. **Artists**: Contains details about music artists, such as their name, genre, and country of origin. This information is essential for analyzing the popularity of artists across regions and genres.
3. **Songs**: Stores information about each song, including its title, artist, album, duration, release date, and genre. This table links to the Artists table and enables us to analyze song details, artist collaborations, and genre trends.
4. **Playlists**: Represents the playlists created by users, with fields for playlist name, creator, and creation date. This table allows us to track user-created playlists, understand playlist creation trends, and analyze playlist preferences.
5. **Playlist\_Songs**: This associative table connects playlists to songs, representing a many-to-many relationship between them. It enables analysis of song popularity within playlists and allows for playlist-specific insights.
6. **Listening\_History**: Captures each instance a user listens to a song, including the timestamp. This table is crucial for understanding song popularity, user engagement, and listening behavior over time.
7. **Subscriptions**: Stores details about each user’s subscription, including subscription type, start date, and end date. This table helps in understanding user retention, subscription trends, and the popularity of different subscription models.

**Example Analysis and Business Questions**

Using this database structure, analysts can answer the following types of questions to generate business insights:

1. **User Demographics and Subscription Patterns**:
   * How many users are on Free vs. Premium subscriptions?
   * What is the average time since a user joined the platform?
   * Which countries have the highest percentage of Premium users?
   * Identify users who haven’t used the platform in the past month.
2. **Artist and Song Popularity**:
   * Which artist has the most songs in the database?
   * Which genre is the most popular based on the number of songs and listening counts?
   * List the top 5 most-played songs across the platform.
   * Track the release years of songs and identify trends in music releases.
3. **Playlist Trends and User Engagement**:
   * Determine the average number of songs per playlist.
   * List playlists containing songs by a particular artist (e.g., Taylor Swift).
   * Identify the playlists with the highest user engagement.
   * Find playlists created by Premium users only, which can help tailor content for these users.
4. **Listening Behavior**:
   * Track which songs are most frequently listened to by users.
   * Find the total number of listens per artist to identify popularity.
   * Identify top users with the highest number of unique songs listened to, useful for user retention analysis.
   * Track daily listening counts for specific songs to understand peak usage times.

**Example Queries**

The SQL queries provided cover various questions such as finding popular songs, identifying user activity, and tracking artist or song popularity. Each query demonstrates how to extract meaningful insights, such as:

* **User Engagement**: Queries analyze user engagement by looking at song listens and playlist activity.
* **Content Preferences**: Queries help determine song or artist popularity across playlists and genres.
* **Subscription Analysis**: Understanding the distribution of Free vs. Premium users provides insights into revenue models.

**Potential Applications**

This project can be expanded or used as a starting point to develop more advanced analytics and insights for a music streaming platform. Here are a few applications:

1. **Recommendation System**: Analyzing listening patterns can help develop personalized song or playlist recommendations.
2. **Marketing Campaigns**: Identifying user demographics and popular artists can tailor marketing campaigns for specific genres, countries, or subscription types.
3. **User Retention Strategies**: Understanding when users are active or inactive can guide retention strategies for Premium users.
4. **Artist and Genre Insights**: Knowing which artists or genres are popular allows better content acquisition and helps promote trending genres.

**Conclusion**

The Spotify Analytics Database is a robust tool for simulating real-world analysis scenarios for a music streaming service. With a well-structured database, analysts can uncover insights about user behavior, content preferences, and subscription patterns. This project demonstrates the use of SQL to transform raw data into actionable insights, which can be applied to drive product development, enhance user experience, and support business growth strategies on a music platform.