Databases Project Proposal: MusicBase

CS-3743-002 Database Systems

The University of Texas at San Antonio

Group 4 - Swag Software

Table of Contents

Overview

The purpose of MusicBase is to design and implement a database system for a music streaming library, similar to spotify, apple music, or pandora. The database will store information about songs, artists, albums, libraries, and users. The system will allow users to create and manage their libraries, listen to tracks, give reviews, and explore music by genre, artist, or album.

We chose this project because music is a universal form of entertainment that anyone can enjoy, and streaming platforms have become the main way people listen to music (including all of us personally). By developing a relational database to model this system, the project will highlight how data can be structured, stored, and retrieved efficiently just like a real world application. MusicBase can also give all of us experience with entity-relationship modeling, SQL, and enforcement of business rules, all which are essential skills.

Entities

MusicBase will contain the following entities:

- User - The individuals that use this platform:
    - user_id
    - name
    - email
    - password
    - Join_date
    - follower_count
    - following_count
- Artist - Represents music artists or bands
    - artist_id
    - artist_name
    - country
    - genre
    - monthly_listeners
    - followers
- Album - Collection of songs by artists
    - album_id
    - album_name
    - release_date
    - artist_id (Foreign Key)
    - album_rating
    - label
    - rating
- Song - Represents tracks on the platform
    - song_id
    - song_name
    - song_duration
    - song_genre
    - release_date
    - album_id (Foreign Key)
    - artist_id (Foreign Key)
    - rating
- Review - Represents a user's review of a song
    - review_id
    - user_id (Foreign Key)

- review_type
- song_id (Foreign Key)
- rating
- review_text
- creation_date
- like_count

Entity/Relationship

- Artists have 0 to many songs
- Artists have 0 to many albums
- Users follow 0 to many users
- Users follow 0 to many artists
- Users make 0 to many reviews
- Users like 0 to many reviews
- Albums have 1 to many songs
- Albums are made by 1 to many artists
- Songs have 1 to many artists
- Songs have 0 to many reviews
- Songs can belong to 0 to many albums
- A review is made by 1 user
- A review has 1 song

Business Rules

User Rules:

- Users must have a unique username
- Users must create account with an email
- Users can write only one review about a song, a new review will replace the old
- Users can follow other users
- Users can leave likes on reviews
- Users must have unique ID's
- Users can delete and edit their reviews

Artist Rules:

- Artists can have many songs
- An artist can be a group/band
- Artists must have unique ID's

- Artists must have a name
- Artists can release singles
- An EP is counted as an album

Review Rules:

- A review can only belong to 1 user and 1 song
- Must have a rating 0-5 stars
- Must have a creation date and update date
- Can only be deleted by the creator of review

Song Rules:

- Must have unique ID
- Must have a title, an artist, release date, and duration
- Can have multiple genres
- Song can belong to many artists (collab song)

Album Rules:

- Albums can have many songs
- Albums must have a title and release date
- Must have at least 1 song
- One album can belong to many artists (collab album)

Data Retrieval

We will use the MusicBrainz API as it has access to lots of data on artists, tracks, playlists, genres, labels, and more giving us a lot of information to work with. This data is given to us in JSON format and can be retrieved through http requests. The data is allowed to be used for non-commercial use so we will be easily able to access data without any legal issues. The only issue we may run into would be rate limiting as we are not allowed to make more than one query per second.

https://musicbrainz.org/doc/MusicBrainz_API

Conclusion

MusicBase is a database designed and modeled to give users information on their favorite artists that can be found on modern music platforms such as Spotify or Apple Music. Our system has key entities and establishes clear relationships between each entity and states clear business rules to govern their interactions. Additionally, our use of the MusicBrainz API allows us to have rich and reliable datasets, allowing our database to be filled with real data. Applying these concepts provides MusicBase with a well structured framework and gives our team valuable experience in database design through a relevant and engaging domain.