CS480 – Course project

Summer 2021

Database: music_manager

Description:

Our database will house a collection of songs. Each song will have a unique ID, name, artist/artists, the number of streams, format, genre, duration, track number, and if applicable an album or mixtape. The albums are going to be recognized by a unique ID, name, total duration, release date, genre, artists, producers, samples, and the number of songs it has used. Every user has a unique username, age, playlists, artists following, albums liked, songs liked, and friends. Each artist and producer will be described by a unique ID, genre, name, and has a list of their albums and songs. The first intention is to give better and faster search results/recommendations. The second intention is to allow a user to discover related music such as samples or music friends listen to. Names of songs, albums, and artists can coincide with each other but will ultimately differ by some unique ID. For example, an artist can have a song that has the same name as another artist/album. The song ID is a combination of characters that include the name, genre, artist’s name, and album name.

Part 2 – CRUD (Create, read, update, and delete)

Deadline: July 17, 2021

List of strong entities:

1. User
2. Song
3. Artist
4. Genre

List of weak entities:

5. Playlist
6. Album
7. Search
8. Following
9. Info

We will implement the following functionality using Java and SQL with necessary GUI interfaces.

1. Insert/delete/update/read a **User** (all attributes except the user id). The member id should be generated by the system automatically using MySQL autoincrement.
2. Insert/delete/update/read a **Genre** (all attributes except the genre id). The genre id should be generated by the system automatically using MySQL autoincrement.
3. Insert/delete/update/read a **Song** (all attributes except the song id). The song id should be generated by the system automatically using MySQL autoincrement.
4. Insert/delete/update/read a **Artist** (all attributes except the artist id). The artist id should be generated by the system automatically using MySQL autoincrement.
5. Insert/delete/update/read a **Playlist** (all attributes except the playlist id). The playlist id should be generated by the system automatically using MySQL autoincrement.
6. Insert/delete/update/read an **Album** (all attributes except the album id). The album id should be generated by the system automatically using MySQL autoincrement.

Part 3 – Queries

Deadline: July 31, 2021

Based on the Demo, we will implement the following functionality using Java and SQL with necessary GUI interfaces.

Trivial Queries:

1. List all users
2. List all genres
3. List all songs
4. List all artists

Non-trivial Queries:

1. List all the friends of the users in an order(alphabetically) by last name
2. List all the common songs that the user shares with his/her friends.
3. List all the albums and artists that both the user and his/her friends like.
4. List more songs/albums of the artist the user and his/her friends listen to the most.
5. List all the playlists of a user in an order.
6. List more songs from a genre that the user and his/her friends listen to the most.
7. List all the playlists of a user’s friends.
8. List the most popular genre from the user’s playlists.