

CS480 – Course project

Summer 2021

Database: `song_chart`

Description:

I am interested in designing a database system for keeping track of all songs that come out from artists, and the top charts of the most popular songs in each week. I was motivated because I love listening to music and want to keep track of all the songs and their properties and see if there are any patterns to the top songs. A collection of songs are collected, each of which has a unique song ID and a song name, The artists who made the song in order of contribution significance, genre of the song, number of streams, genre, album name of the associated song, and the BPM of the song. The artists who made the song are also collected, each of which have a unique artist ID, name, nationality, how many albums released, number of songs released and awards. The albums of each artist are also collected with a unique albumId, and the name of the album is collected with the name of the artist, date of release, length of album, album rating, and total streams from it. Each top chart consists of an associated date of the chart, the song name, the song id, artist name, artist ID, the BPM of the song, the genre of the song, and the weeks the song has charted.

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

Deadline: July 17, 2021

List of strong entities:

1. artist
2. album
3. song

List of weak entities:

4. artist_name
5. song_name
6. album_name
7. week_chart

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

1. Insert/delete/update/read an **artist** (all attributes except the artist id). The artist id should be generated by the system automatically using MySQL autoincrement.
2. 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.
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 **week_chart** (all attributes). The dates will have to be manually put in and attributes put in for each week.

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 artists
2. List all songs
3. List all albums

Non-trivial Queries:

1. List all the songs by a particular artist with a specified bpm
2. List all the songs in which the streams are over a billion with a specified genre.
3. Find all the albums with a rating over four stars with the lowest streams.
4. List the songs that have been in the top charts the longest; if there is a tie, list all songs that are part of the tie.
5. Define a VIEW called fastSongs and then display all the songs with the highest bpm that have been in the top charts.
6. Find the average streams of songs by genre
7. average rating of albums with particular bpm
8. find the average bpm of songs in the top charts by year