CPSC 2221 - Database Systems
Group Project - Implementation of a Relational Database

Project Title:	Community Music Database
Project Milestone:	#4 - Complete Project Files

#	Student Name	Student ID	Email Address
1	Chrissie Koles	100346764	kkles@mylangara.ca
2	Teresa Alves	100350471	talvesesilva00@mylangara.ca
3	Kathryn Vit	100337651	kvit00@mylangara.ca

By keying our names and student IDs in the above table, we certify that the work submitted with this cover page was performed solely by those whose names and student IDs are included above.

Also, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Langara College.

This project consists in a database created in MySQL. It also provides a web GUI interface that can be used to check specific queries with data within the database. The GUI was programmed using HTML and CSS(Bootstrap).

Server name: localhost
Username: root
Password: root
Database: musicdb

The components and the minimum requirements are:

- XAMPP v3.2.4
- Web Browser

Application installation instructions:

1. After installing XAMPP, run the XAMPP Control Panel. Start Apache and MySQL services.
2. Extract this repository to /xampp/htdocs directory or change the root folder in \xampp\apache\conf\httpd.conf to the desired directory.
3. Go to https://localhost/phpmyadmin/ and create a new database called musicdb
4. Go to https://localhost/phpmyadmin/server_privileges.php?db=musicdb and change the password of root @ localhost to “root”.
5. Go to https://localhost/phpmyadmin/db_sql.php?db=musicdb and run the contents of the CREATE_Script_MusicDB.sql file.
6. In the same page, run the contents of the INSERT_Script_MusicDB.sql file.
7. Visit http://localhost/MusicDB_Project/src/ and test the queries.

Short description of what the project accomplished:

This project is a music database where users can search for artists, songs, albums, etc. It is a place where audiophiles and artists alike can come together to search for music and add music to an ever-growing database of information.

This Community Music Database (CMD) does not hold the .mp3 files, but rather information about the music. What songs has a particular artist released? Who is the artist of that catchy song you can't get out of your head? Is there any music I haven't heard yet from my favourite musician? These are the kinds of questions we seek to help users answer with our CMD.

Our goal is to have a neatly organized database that can grow with its users - after all, musicians who have the ability to adapt to changing sounds have more longevity than those who do not. We hope our CMD will have a long life like an adaptable and well-rounded musician.

List of the SQL queries used:

1. Projection Query:

```
SELECT artistID, artistName FROM artist;
```

2. Selection Query:

```
SELECT artistID, artistName FROM artist WHERE artistName = '$SArtist';
```

3. Join query:

```
SELECT artist.artistName, album.albumTitle, album.releaseYear
FROM artist
INNER JOIN artistAlbums ON artist.artistID=artistalbums.artistID
INNER JOIN album ON album.albumID=artistalbums.albumId
WHERE artistName = '$SArtist';
```

4. Division query:

```
SELECT tagName
FROM tag AS t
WHERE NOT EXISTS(
(SELECT a.albumId
FROM album AS a)
EXCEPT
(SELECT atags.albumId FROM albumtags AS atags WHERE atags.tagId =
t.tagId));
```

5. Aggregation query:

```
SELECT MIN(releaseYear)
FROM Album;

SELECT MAX(songNumber)
FROM albumsongs;
```

6. Nested Aggregation with Group-By:

```
SELECT AVG(songs.max)
FROM ( SELECT MAX(songNumber)as max FROM albumsongs group by
albumId)songs;
```

7. Delete Operation:

```
DELETE FROM artist WHERE artistname = '$AName';
```

8. Update Operation:

```
UPDATE album
SET releaseYear = '$NewYear'
WHERE albumTitle = '$AName';
```

9. Extra features

   Our GUI is responsive and beautifully designed with Bootstrap.