INFO 210: Database Management Systems

Term Final Project

Hanae Ferahi, Sahithi Pisupati, Makahra Gayle, Shavan Patel

December11, 2019

Project Category: Oracle Implementation

Title: Design and Implementation of a Multimedia Music Database System

Part A: Proposal

1. The Problem Statement

(a) Overall goals of the system

A music multimedia database system in any application area such as images, photos, audio data, songs, or video clips for any applications related to music.

(a) Context and Importance of the system

There was no previous multimedia system in place for the stakeholder. It is crucial that a music multimedia database system can include graphics, text, images, video etc. As this is a multimedia database system, it should allow multimedia data to be modified and stored. In general, a multimedia database system can keep track of items for purpose such as documents, record management, education, training and knowledge dissemination [1]. It will also keep track of these items for Human Resource Applications for keep tracking of employee, music databases, and travel agencies to keep track of attractions.

(b) Scope of the project:

IN-Scope:

Music multimedia database system will include images, photos, audio, video etc. for only clients and their dependents. This includes data regarding singers, composers, CDs, song etc.

OUT-Scope:

Music Multimedia database system will **not** include other non-music-related activities such as movies, TV shows etc.

2. Requirements

2.1 Data requirements

For Music Multimedia Database, it must provide an efficient storage and manipulation of multimedia data. On the other word, we should keep track of Music database keeping track of singers, album, and CD images, and piece of their songs. In addition to the integration of large amounts of complex structured data in order to have a developed DBMS technology.

For each singer, DBMS will need to keep track of artist-ID, artist name, total number of songs, total number of albums, and image ID.

For each album, DBMS will need to keep track of the album ID, name, length, the released date, total tracks in album, the genre, the image associated with the album, and most importantly is download permit.

For each CD image, DBMS will need to keep track of image ID and image location.

For each Song, DBMS will need to keep track of song ID, title of song, length of the song, the album ID if it is belonging to an album, the genre of the music, the download permit Boolean and the shared link.arti

2.2 Business Rules and Data Logic

(1) You can find a song through a name of the artist, song, or album
(2) Every song must have an album name and artist name
(3) When an album is searched for within the database, a link to its image will appear.
(4) Along with the CD image, the database will also show the number of songs within the album, the total length of time for all the songs, all the songs within the album, and the year the album was made.
(5) When a song is searched for or selected within the database, its length of the song will be shown, as well as the genre, year, and artist name.
(6) When an artist is searched for within the database, all his/her albums will be linked for the user to then be able to select specific albums
(7) The artist's genre(s) of music will also be shown when the user selects or searches for an artist

2.3 Sample Output

(1) Display the song with the appropriate image associated along with the music information such as the compositor, distributor, and artist name.
(2) Display the numbers of likes and shares associated with the songs.
(3) Generate a statistical report of the views and downloads based on the selected time frame.
(4) Output all the songs that is belonging to the same category such as album, genre, artist, etc.

2.4 Other Assumptions

(1) We will assume DBMS will be used by a record-producing label that operates in the music industry
(2) DBMS runs on a server environment, running Windows Server as OS.
(3) The underlying DBMS system is Oracle.

3. Project Plan

The need for a relational database was determined by stakeholder management. Requirements were sourced from internal staff as well as other similar products such as the online music database like MusicBrainz [2].

Expected project timeline is about 7 weeks. Project will be managed and developed via SCRUM methodology. SCRUM will allow for the most efficient development process. The multimedia databased will be hosted via Oracle in a Windows 10 environment. Ongoing management and support will be provided by a collaboration between operation and development team. There is no previous database in place so there is no need for a parallel startup. This is commonly known as DevOps which simplifies and encourages collaboration between the development team and operations department to increase customer satisfaction from.

4. Proposed Deliverables and Division of Work

Our final deliverables will consist of an analysis of a current music database system, a database schema and an ERD model of the music database system. We will be providing an example of a current music database and the pros and cons of that system. Then, we will use this information to generate a database schema and an ERD model of the system of our own. The database schema will show the links/relationship between the tables in our database and the ERD will be used to establish the links between the entities of our system, which will be used to offer a visual presentation of our system layout.

We will then implement our database into SQL. The SQL implementation will contain queries that will show the database's ability to properly store/modify the data. The final

deliverable that will be submitted is a summary that will go over what we were able to achieve, any major bugs or limitations that we ran into, what we would do if more time were permitted, and anything we would've done differently if we could do the assignment again.

As for the division of work, our plan is to divide half the work while doing the other half together. The analysis of a current music database will be done together so we can all gather a proper and similar understanding of the ins and outs of the current system. From there, we will then split off and work on developing our own ERDs for the system. Once done, we will meet up and work on implementing the best features of our individual ERD into one main ERD that will be used for the project. Once we have an ERD established, we will work together to create a database schema of our music database system. Once the ERD and database schema are approved by the professor, we will then work together on implementing the database in Oracle. Once properly implemented, we will individually work on SQL queries that show the effectiveness of our database, and we will each document our queries in our final deliverable.

At the end of the project, once everything is established and completed, we will compress our files into a single RAR file. This file will then be uploaded to Blackboard before it is due.

5. Known References

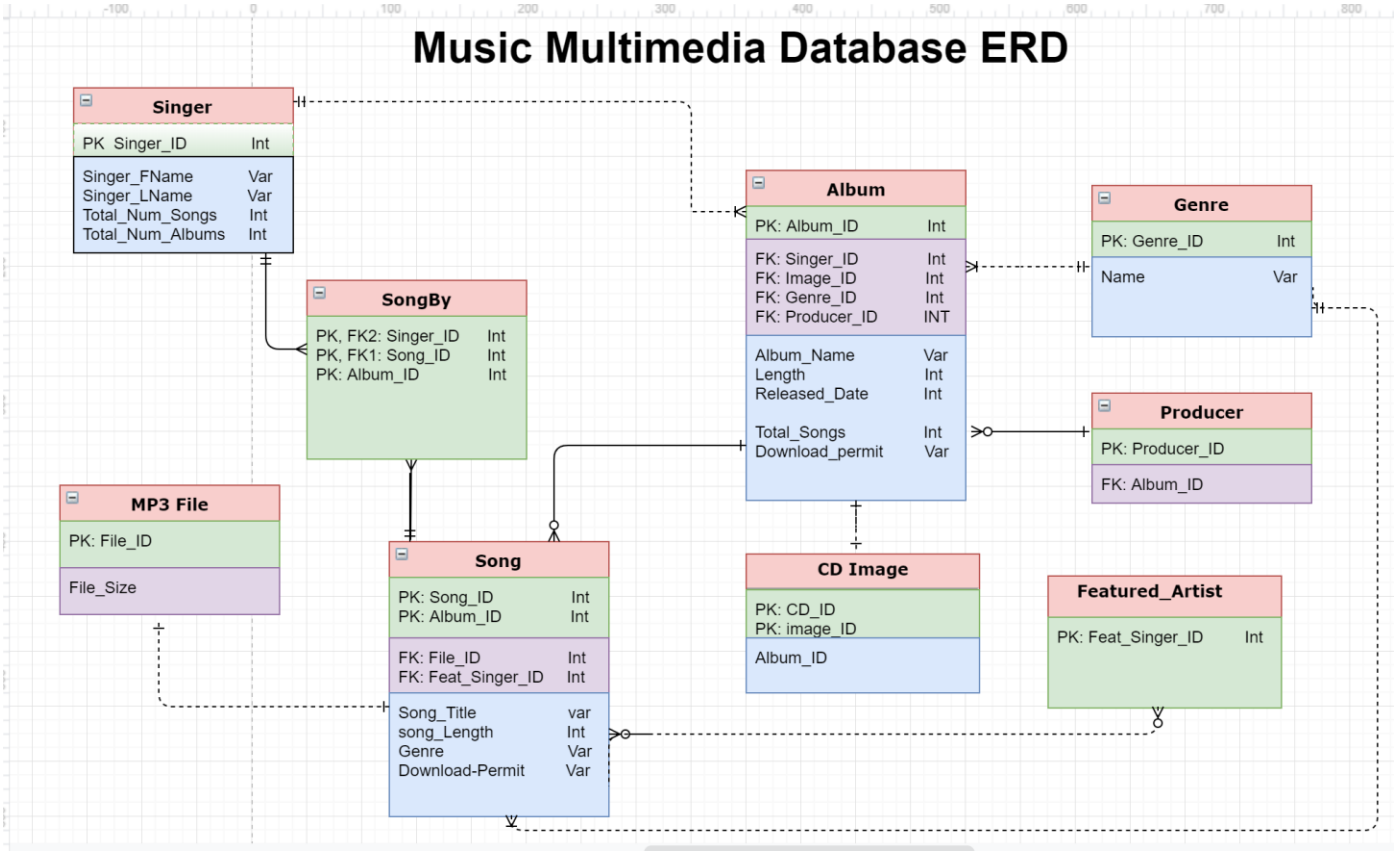
[1]. Singh, Himanshi. "Multimedia Database." *GeeksforGeeks*, 20 Aug. 2019, `www.geeksforgeeks.org/multimedia-database/

[2]. "Database Statistics." *MusicBrainz*, musicbrainz.org/statistics.

https://musicbrainz.org/statistics

Part B: ERD and Relational Schema

ERD



RDB Schema

CD Image (**CD_id**, Image_ID, Album_ID)

FOREIGN KEY Album_ID REFERENCES Album(Album_ID)

Song (**Song_ID**, **Album_ID**,Singer_ID, File_ID, Song_Title, song_Length, Genre, Download-Permit)

FOREIGN KEY File_ID REFERENCES MP3 File(File_ID)

Album (**Album_ID**, Singer_ID, Image_ID, Genre_ID, Producer_ID ,Album_Name, Length, Released_Date, Total_Songs, , Download_permit)

FOREIGN KEY Singer_ID REFERENCES Singer(Singer_ID)

FOREIGN KEY CD_ID REFERENCES CD Image(CD_ID)

FOREIGN KEY Genre_ID REFERENCES Genre(Genre_ID)

FOREIGN KEY Producer_ID REFERENCES Producer(Producer_ID)

Singer (**Singer_ID**, Singer_FName, Singer_LName, Total_Num_Albums Total_Num_Songs, Total_Num_Albums)

Featured_Artist (**Feat_singer_ID**,Singer_ID)

Producer (**Producer_ID,** Album_ID)

FOREIGN KEY Album_ID REFERENCES Album(Album_ID)

MP3 File **(File_ID** , file_size)

Genre(**Genre_ID**, name)

SongBy (**Song_ID, Album_ID ,Singer_ID**)

FOREIGN KEY Singer_ID REFERENCES Singer(Singer_ID)

FOREIGN KEY Song_ID REFERENCES Song(Song_ID)

Part C:

The team decides the sharing of workloads of creating tables and inserting the data into the tables (who will create which tables and inserts what data to which tables.)

• Each member creates CREAT TABLE commands and INSERT commands of assigned tables. Note that data to be inserted must be agreed in advance among members due to the consistency between PKs and FKs.

• A member collects them in one file and test them to see whether they all work fine. Then, the member shares the DDL and INSERT among members.

• Once the complete DDL and INSERT works, then members get together and decides what queries and what DML commands to create for each member. The queries should be meaningful and complex enough ones that contain at least two of the following a) multi-table joins, b) aggregate functions; c) several WHERE conditions on date or string fields, etc.

• Each member creates the whole database in each member's account. • Each member can create their own queries and DML commands and captures the output, including SQL commands, output, and #rows retrieved.

• Each member displays all the data of assigned tables using SELECT * command. • A member collects all the SQL and DML outputs.

• A member creates the final report with all the required components, including problem statement, ERD, CREATE TABLE commands, INSERT commands, SQL commands and output, DML commands and output, and any final document. The documentation effort must be shared among team members.

• Then, the final report that consolidates all the work is submitted by one member to Assignment 10 folder of Week 10.

Data Collection: