Authors: Joshua Patterson, Alex Richardson, Justin Wiggins

November 6, 2019

CMSI 486

Detailed Database Design

1.1 – Project description, database engine used, potential users, maybe some other stuff

For our project, we will create a music song database using mySQL. This will organize each song by genre, sub-genre, arist, and bpm. This database will work hand-in-hand with our 401 project, where real song samples are used to train new models using Generative Adversarial Networks. These song samples will be allocated within the database, and be ordered by their specific “tags” (artist, genre, sub-genre, bpm). Potential users of this database could be those who want to discover songs with similar bpm/genre/artists for music production or for music implementation within neural networks.

1.2 – Data description, generally what type of data will be stored

We will largely store string data, both categorical and identifiers. There will be a lot of overlap in genre, subgenre, and artist categories. A stronger identifier will be song name, though not necessarily unique. We will also store either wav or mp3 files. Since those two types of files are distinct formats, we have to decide if we want to prioritize smaller files or higher quality audio.

1.3 – At least five examples of the type of data your database will provide *to the user*

1. Allowing the user to find a given artist and their songs that are utilized for our neural network
2. Allowing the user to search a given genre (hip-hop, lo-fi, r&b, etc..) and find artists/songs that share the same genre.
3. Allowing the user to search a given sub-genre (hip-hop, lofi, r&b, etc..) and find artists/songs that share the same sub-genre.
4. Allowing the user to search a given BPM and to find songs with the same/similar BPM.

5. Allowing the user to search a given song that is utilized within our neural network, and to see the song's "tags" (genre, sub-genre, bpm, artist).

1.4 – A preliminary idea of the schema of the database including table descriptions and potential columns

The database will have two entities. The first entity/table will be the artist table, which will have attributes such as the artist name, artist genre, and song name. The second entity/table will be song information. This will have attributes such as song bpm, song genre and song sub-genre.

You must expand your Preliminary Design *in a new copy of the document* to have the following as part of your detailed design:

- **2.1 – Project description, database engine used, potential users, maybe some other stuff**

As Explained in our Preliminary Design, we are creating a song database for the samples used for our Generative Adversarial Network. This will organize each song by genre, sub-genre, arist, and bpm. This database will work hand-in-hand with our 401 project (yung gan), where the songs within this database are used to train new models through a neural network. These song samples will be allocated within the database, and be ordered by their specific "tags" (arist, genre, sub-genre, bpm). Users of this database will be those who want to find the specific songs within our database with similar artists, bpm's genres and sub-genres for music production/ implementation through Generative Adversarial networks. We wanted to expand upon our Preliminary Design by adding an entity entitled "Song Interaction", in which the user can grab information from this table and implement it directly within their projects.

- **2.2 – Data Dictionary, meaning a bullet list of the final tables/columns with a complete description of each**

Entities:

1. Song Information
    a. Attributes Include : Song Artist, Song Genre, Song Sub-Genre, Song BPM
        i. These attributes describe the basic information about each song, in which users could search specific attributes and find

overlaps between multiple songs/artists/genres/sub-genres and BPM.

2. Song Interaction
    a. Attributes Include : Song Index Number, Song MP3 Files, Song Wav File
        i. These attributes describe both the specific index we have given each song in our database along with a WAV and MP3 file for each song. These indexes simplify our way of searching for songs we want to utilize within our neural network, and which songs need to be weeded out to generate more precise models. Examples would be songs with lyrics, excessive breaks and unnecessary interludes. Having MP3 and WAV files will efficiently speed up the process of using these songs as samples for music production or for neural network training.

- **2.3 – A complete finalized Entity-Relationship Diagram [ERD] for the database [NOT hand-drawn, PLEASE!]**