CIS 550 Group Project Outline

Drive link: https://docs.google.com/document/d/1hYCUbSAyJmWUl-f3tVaeS5iObP5gEbj4dgVGp0Do\_rk/edit#heading=h.5xeubomnkljf

**Motivation for the idea/description of the problem the application solves**

The website with Spotify music collections aims to provide an interactive interface with multiple entries for users to search Spotify songs and add those songs to their own playlist once those songs are in their favor.

**List of features you will definitely implement in the application**

1. allow users to search songs with specific filters (keywords of song names, artist)
2. allow users to search songs with unspecific filters (country, year, emotional tendency)
3. allow user to add/remove songs to playlist

**List of features you might implement in the application, given enough time**

1. A detailed breakdown of parameters of emotional tendency for more accurate filter
   1. Stirring
   2. Empowering
   3. Fiery
   4. Sensual
   5. Lively
2. Add sub-filters (live search box) in page 1

**List of pages the application will have and a 1-2 sentence description of each page. We expect that the functionality of each page will be meaningfully different than the functionality of the other pages.**

1. One page with 3 features:
   1. *Country Filter*. When users select to click the button of Country, an interactive map will appear on the webpage. Users are able to click one of countries, and then a list of songs generated by artists in that country will pop up
   2. *Year Filter*. When users select to click the button of Decade, a sub-list of decades (60s and before, 70s, 80s, 90s, 00s, 10s, 20s and after) will present to allow users to select which decade the songs are in.
   3. *Emotional Tendency*. When users select to click the button of Emotional Tendency, they can choose whether to list happy songs or sad songs.
2. One page with search functionality to allow users to search songs via:
   1. Song name
   2. Artist name
3. One page with a playlist for each individual user, and users are able to add songs to the playlist from 1 and 2, as well as remove songs on this page.

**Relational schema as an ER diagram**

See Appendix A.

**SQL DDL for creating the database**

See Appendix B.

**Explanation of how you will clean and pre-process the data. This tutorial demonstrates how to do simple pre-processing in Python.**

**EDA:**

1. We downloaded the dataset from Kaggle: *Spotify music analysis*, and then we used web scraper script to pull out other song attributes such as song name, genre, artist name, artist country, and lyrics snippet. As well as artist album and year etc.

2. Check dataset measurement, missing data, distribution and duplicates. Pandas.info(), Pandas.describe(),and Pandas.plot() are able to suffice our needs.

**Pre-process:**

Song Features

1. We fix the column name and change the ‘h/s’ column to ‘h\_s’, and ‘key’ to ‘key\_pitch’ to circumvent MySQL constraints.

2. Remove duplicates: we use Pandas.drop\_duplicates() to remove duplicated

3. After dropping the duplicates, we still found out that duplicate ‘uri’ exist as there are different ‘h\_s’ values and ‘danceability’ for one ‘uri’. In order to keep ‘uri’ unique, we further adjust to remove duplicate values for uri, we keep ‘h’ value for duplicate ‘h\_s’ and larger ‘danceability’.

4. Double check the dataset info and convert to csv.

Artist

1. For the Album year, we will extract the ‘year’ from release\_date, the original formats of which vary from ‘DD-MM-YYYY’, ‘MM-DD-YYYY’ to ‘YYYY’.
2. For the missing value, we will replace with N/A

Song Details

1. Remove the unused column ‘snippet’
2. Replace other missing value with N/A

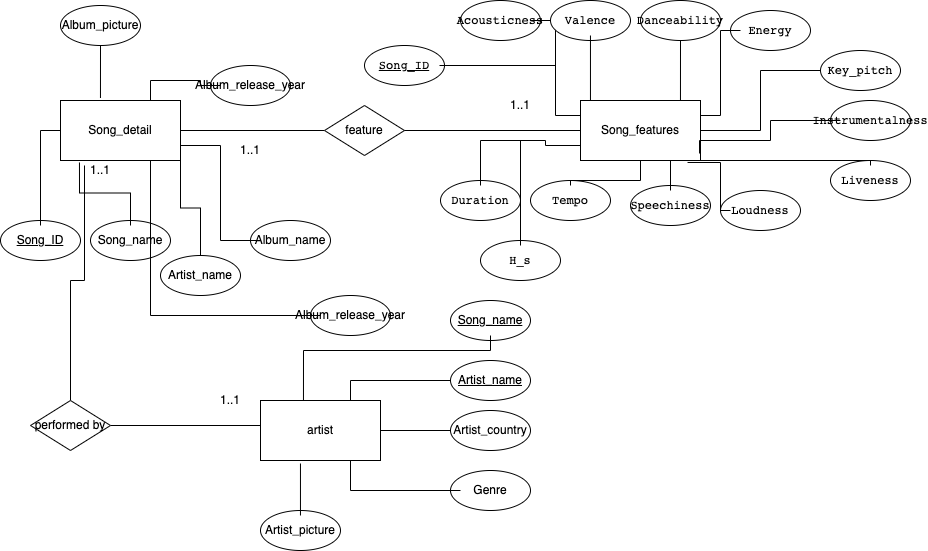
**List of technologies you will use. You must use some kind of SQL database. We recommend using MySQL specifically because you will use MySQL in HW2, and we will provide guidance for setting up a MySQL database.**

1. Python:
   1. Pulling spotify data using a third-party tool called spotipy, which uses spotify api.
   2. Pulling supplementary data from Musixmatch using musixmatch api.
   3. Data pre-processing such as cleaning, integration, reduction, and transformation using the panda library mentioned above
2. Excel:
   1. Raw data storage
   2. Data preprocessing before importing into the database
3. MySQL: Using MySQL hosted on AWS RDS as our database since all relations in our dataset are well defined
4. JavaScript /HTML /CSS: Website design
5. FIFA-themed web application template using React and Node.js to create our website

**Description of what each group member will be responsible for**

* Huaying – dataset and metadata preparation, relational schema (ER diagram) generation
* Fonda – data preprocessing, data importing, SQL DDL generation
* Mingwei – data preprocessing
* Haoning – Web page prototyping, project documentation

**Appendix A: Entity Relationship Diagram**



**Appendix B: SQL DDL**

CREATE DATABASE Songs\_artist;

USE Songs\_artist;

CREATE TABLE Song\_features(

Song\_ID varchar(100),

Acousticness DECIMAL(8,8),

Valence DECIMAL(4,4),

Danceability DECIMAL(4,4),

Energy DECIMAL(6,6),

Instrumentalness DECIMAL(8,8),

Liveness DECIMAL(4,4),

Loudness DECIMAL(10,3),

Speechiness DECIMAL(6,6),

Tempo DECIMAL(10,3),

Duration INT,

Key\_pitch INT,

H\_s varchar(1),

PRIMARY KEY (Song\_ID)

);

CREATE TABLE Song\_detail(

Song\_ID varchar(100),

Song\_name varchar(100),

Artist\_name varchar(50),

Album\_name varchar(50),

Album\_picture varchar(225),

Album\_release\_year INT,

PRIMARY KEY (Song\_ID),

FOREIGN KEY (Song\_ID) REFERENCES Song\_features(Song\_ID)

);

CREATE TABLE Artist(

Song\_name varchar(100),

Genre varchar(100),

Artist\_name varchar(50),

Artist\_country varchar(50),

Artist\_picture varchar(255),

PRIMARY KEY (Song\_name,Artist\_name),

FOREIGN KEY (Song\_name) REFERENCES Song\_detail(Song\_name),

FOREIGN KEY (Artist\_name) REFERENCES Song\_detail(Artist\_name)

);