MUSIC RECOMMENDATION DATABASE
PROF. Nick Brown

Contributors:
Abhishek Hegde [NU ID: 002744522]
Lokesh Jeswani [NU ID: 002795957]
Team: "Music Junkies"

Abstract:

A Music Database is created with carefully chosen entities such as songs, albums, genres, artists, and record labels. The project has a command-line interface that offers database capabilities like adding and deleting data, altering values, sorting by an attribute, building views, and performing various queries. Choose any song based on your preference or mood. The database will display a playlist, or a list of recommended songs based on the genre, song or artist chosen. Here, we've taken datasets from the "Spotify" music service, kept the pertinent entities that the database schema required, and imported them.

Introduction:

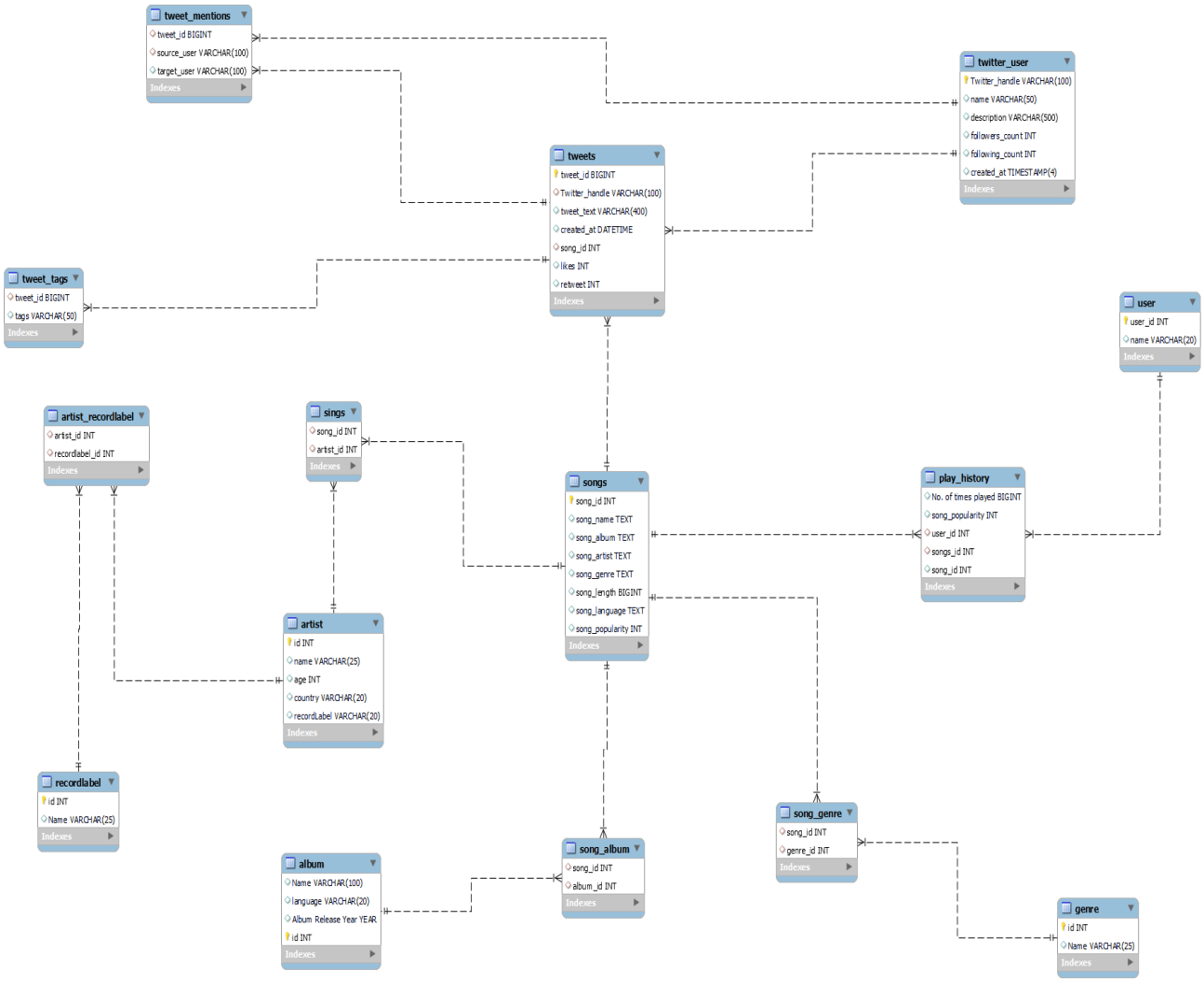
Many music listeners have turned to listen to online music. The database technology has made it possible that music listeners could get access to music as they want. Online service of music subscription has gained immense popularity in the era of cloud computing. The advancement of cloud techniques eases users to get access to an unlimited number of songs. Here we will be creating "Music Recommendation Database", which will provide recommendations based on the genre and various other parameters of the song, which is currently being played.

Music Recommendation System is used to recommend songs based on genre. The system allows users to create playlists based on genre and popularity whenever they are logged in. Recommendations are also made based on genre and various other relational entities. The system can provide instantaneous recommendations for each song which is being played.

Use-case for un-authorized users:
1. Search songs and albums through database using input keywords.
2. View recently added albums and its details.
3. View custom music playlist and get recommendations based on it.

Use-case for authorized users(admin):
1. Can add and remove albums from the playlists and database.
2. Add new music/albums to the database.
3. The currently authorized user or administrator can see the updates to the uploaded data.

ER Diagram:



Snippets from Database created:

Songs Table:

song_id	song_name	song_album	song_artist	song_genre	song_length	song_language	song_popularity
1	Beat It	Thriller 25 Super Deluxe Edition	Michael Jackson	Pop	3	English	81
2	Smooth Criminal - 2012 Remaster	Bad 25th Anniversary	Michael Jackson	Pop	3	English	79
3	Gimme More	Blackout	Britney Spears	Pop	3	English	78
4	Eye of the Tiger	Rocky IV	Survivor	Rock	3	English	76
5	Everybody Wants To Rule The World	Songs From The Big Chair (Super Deluxe Edition)	Tears For Fears	R&B	3	English	86
6	Everybody Talks	Picture Show	Neon Trees	R&B	2	English	79
7	Earned It (Fifty Shades Of Grey)	Beauty Behind The Madness	The Weeknd	Pop	3	English	74
8	In Da Club	Get Rich Or Die Tryin'	50 Cent	Hip-Hop	2	English	82
9	Black and Yellow	Rolling Papers	Wiz Khalifa	Rap	3	English	75
10	Superman	The Eminem Show	Eminem, Dina Rae	Hip-Hop	4	English	0
11	In Da Club	Get Rich Or Die Tryin'	50 Cent	Hip-Hop	2	English	75

Album Table:

Name	language	Release_year	id
Thriller 25 Super Deluxe Edition	English	1982	1
Bad 25th Anniversary	English	1987	2
Blackout	English	2007	3
Rocky IV	English	1905	4
Songs From The Big Chair (Super Deluxe Edition)	English	1985	5
Picture Show	English	2012	6
Beauty Behind The Madness	English	2015	7
Get Rich Or Die Tryin'	English	2003	8
Rolling Papers	English	2011	9
The Eminem Show	English	2002	10
Get Rich Or Die Tryin'	English	2003	11

Artist Table:

id	name	age	country	recordLabel
1	Michael Jackson	50	USA	Sony
2	Britney Spears	30	USA	Warner
4	Tears For Fears	34	Ireland	CBS
5	Neon Trees	23	Germany	Universal
6	The Weeknd	35	USA	BMG
7	50 Cent	36	Mexico	ARMIND
8	Wiz Khalifa	32	USA	Sony
9	Eminem	33	USA	Warner
10	C. SHIROCK	28	Ireland	CBS
11	Post Malone	25	USA	Universal
12	DJ Snake	26	USA	BMG

Genre Table:

id	Name
1	Pop
2	Rock
3	R&B
4	Hip-Hop
5	Rap
6	Electronic
7	Funk
8	Bollywood
9	Anthem
10	House
11	Club

Played_history Table:

Played_count	song_popularity	song_id
4680	81	1
2055	79	2
2397	78	3
7647	76	4
9455	86	5
2635	78	6
2554	74	7
7498	82	8
7111	75	9
2652	75	11
1283	80	12

RecordLabel Table:

id	Name
1	Sony
2	Warner
3	CBS
4	Universal
5	BMG
6	ARMIND
7	T-series
8	RCA
9	Epic
10	Atlantic
11	Republic

Below are the tables created using foreign keys of two tables.

Sings Table – shows which artist is the contributor to which song.

song_id	artist_id
1	1
2	1
3	2
5	4
6	5
7	6
8	7
9	8
11	7
13	10
17	14

- *One artist can have many songs, for ex, artist with id 1 have sung songs with id 1 and 2.*

Song-album Table – [shows which song belongs to which album].

song_id	album_id
1	1
39	2
38	2
2	2
3	3
4	4
5	5
6	6
7	7
76	8
11	8

- *One album can have multiple songs in it, for ex, album_id 2 has song_id's 2,38 and 39.*

Artist_recordlabel Table - [connects artist and recordlabel tables]

artist_id	recordlabel_id
1	1
2	2
4	3
5	4
6	5
7	6
8	1
9	2
10	3
11	4
12	5

- *Each artist can have songs released under multiple record labels.*

SNAPSHOT OF VIEWS AND USE CASES:

1) List few songs and its artist: Eminem with album name 'Recovery' and release year '2010'.

```
CREATE VIEW Eminem_2010 AS
select distinct a.song_name, a.song_album, b.name, c.Release_year from songs a
JOIN artist b on b.name = a.song_artist
JOIN album c on c.name = a.song_album
where b.name = "Eminem" and c.Release_year="2010";
```

select * from Eminem_2010;

	song_name	song_album	name	Release_year
▶	Not Afraid	Recovery	Eminem	2010

2) Display few artists and their age whose name starts with 'L' and age is above 40?

```
CREATE VIEW L40 AS
select distinct a.name, a.age from artist a Join songs b
on a.name = b.song_artist
where a.name like 'L%' and age>40;
```

select * from L40;

	name	age
▶	Laura Branigan	67
	Lionel Richie	48

3) Show some popular bollywood songs sung by 'Arijit Singh' with least popularity ratings.

```
CREATE VIEW less_popular_arijit_singh_songs AS select distinct
a.song_name,a.song_artist,a.song_popularity from songs a join artist b on b.name =
a.song_artist join album c on c.name = a.song_album where b.name = 'Arijit Singh' and
a.song_genre = 'Bollywood' order by song_popularity ASC;
```

Select * from less_popular_arijit_singh_songs;

	song_name	song_artist	song_popularity
▶	Palat - Tera Hero Idhar Hai	Arijit Singh	50
	Kabhi Jo Baadal Barse	Arijit Singh	59

4) List the songs with the least number of tweets between 2022-11-1 between 2022-11-12.

```
CREATE VIEW least_tweet_songs AS
Select m.song_name, count(t.song_id) as count from songs m inner join tweets t on
t.song_id=m.song_id group by m.song_name
order by count limit 1;
```

```
select * from least_tweet_songs;
```

Result Grid

song_name	count
Africa	9

5) What are the top 10 songs with highest popular ratings and in English language?

```
CREATE VIEW top10_popular_songs AS
select song_name,song_popularity from music.songs
where song_language = 'english' order by song_popularity desc limit 10;
```

```
select * from top10_popular_songs;
```

song_name	song_popularity
Everybody Wants To Rule The World	86
rockstar (feat. 21 Savage)	84
Viva La Vida	84
Africa	82
Love The Way You Lie	82
Sweet Dreams (Are Made of This) - Remastered	82
In Da Club	82
Love The Way You Lie	82
In Da Club	82
Beat It	81

6) How many unique artists are there in the database where genre is 'Pop'?

```
CREATE VIEW pop_genre_artists AS
select distinct name from artist a JOIN songs b
on a.name = b.song_artist where song_genre = 'Pop';
```

select * from pop_genre_artists;

name
Michael Jackson
Britney Spears
The Weeknd
Dua Lipa
Laura Branigan
TOTO
George Michael
Whitney Houston
Madonna
Ricky Martin

7) Name few artists whose name starts with 'M' and age is above 30?

CREATE VIEW artistM30 AS
select distinct a.name from artist a join songs b on a.name = b.song_artist
where a.name like 'M%' and age>30;

select * from artistM30;

name
Michael Jackson
Madonna

8) List songs with rating above 80 and release year between 2000 and 2015.

CREATE VIEW high_rating80 AS
select a.song_name from songs a JOIN album b
on a.song_album = b.name where song_popularity > 80
and b.Release_year between '2000' and '2015';

select * from high_rating80;

song_name
In Da Club
Love The Way You Lie
Yeah! (feat. Lil Jon & Ludacris)
Toxic
Promiscuous
Viva La Vida

9) How many unique albums does artist 'Michael Jackson' have in the database?

```
CREATE VIEW mj_unique_albums AS
select count (distinct song_album) as unique_albums from songs
where song_artist = 'Michael Jackson';

select * from mj_unique_albums;
```

	unique_albums
▶	3

10) Return songs sung by Arijit Singh with record label 'T-series' and release year between 2013 and 2015.

```
CREATE VIEW arijit_singh AS
select a.song_name, b.recordLabel, c.Release_year from songs a JOIN artist b
on b.name = a.song_artist JOIN album c on c.name = a.song_album
where b.name = 'Arijit Singh' and b.recordLabel = 'T-series'
and c.Release_year between '2013' and '2015';

select * from Arijit_singh;
```

	song_name	recordLabel	Release_year
▶	Kabhi Jo Baadal Barse	T-series	2013
	Palat - Tera Hero Idhar Hai	T-series	2014

11) Show all songs in the database where Eminem has been a contributor.

```
CREATE VIEW eminem_songs AS
select distinct song_name as unique_contributions from songs
where song_artist like '%Eminem%';

select * from Eminem_songs;
```

	unique_contributions
▶	Superman
	Not Afraid
	Love The Way You Lie
	The Monster
	Venom - Music From The Motion Picture
	Rap God

12) List unique songs with rating less than 80 and release year is 1987.

```
CREATE VIEW 1987_80less AS
select distinct a.song_name, b.release_year from songs a JOIN album b
on a.song_album = b.name
where song_popularity < 80 and b.Release_year = '1987';

select * from 1987_80less;
```

song_name	release_year
Smooth Criminal - 2012 Remaster	1987
Man in the Mirror - 2012 Remaster	1987
Dirty Diana - 2012 Remaster	1987

13) What are the top 10 songs with highest popular ratings and artist name starts with 'M' and in English language?

```
CREATE VIEW 10m_popular_songs AS select song_name, song_popularity, song_artist from music.songs where song_language = 'english' and song_artist like 'M%' order by song_popularity desc limit 10;

Select * from 10m_popular_songs;
```

song_name	song_popularity	song_artist
Beat It	81	Michael Jackson
Smooth Criminal - 2012 Remaster	79	Michael Jackson
La Isla Bonita	73	Madonna
4 Minutes (feat. Justin Timberlake & Timbaland)	70	Madonna, Justin Timberlake, Timbaland
Man in the Mirror - 2012 Remaster	69	Michael Jackson
Billie Jean	63	Michael Jackson
Beat It	61	Michael Jackson

14) List few songs and its artist: Michael Jackson with album name 'Thriller' and release year '1982'.

```
CREATE VIEW MJ1982Thriller AS
select distinct a.song_name, a.song_artist, a.song_album, c.Release_year from songs a
JOIN artist b on b.name = a.song_artist
JOIN album c on c.name=a.song_album
where b.name = 'Michael Jackson' and c.name = 'Thriller' and c.Release_year = '1982';

select * from MJ1982Thriller;
```

song_name	song_artist	song_album	Release_year
Billie Jean	Michael Jackson	Thriller	1982
Beat It	Michael Jackson	Thriller	1982

15) What are the top 5 artists of all time, whose song_language is english and genre = 'Pop' based on highest popular ratings?

```
CREATE VIEW top5_artists AS
select distinct song_artist from music.songs
where song_language = 'English' and song_genre = 'Pop'
order by song_popularity desc limit 5;
```

Select * from top5_artists;

song_artist
Coldplay
Eurythmics, Annie Lennox, Dave Stewart
TOTO
Michael Jackson
Nelly Furtado, Timbaland

16) Display song name and its popularity where the number of times song played is greater than 9000.

```
CREATE VIEW most_played AS
Select distinct a.song_name, a.song_popularity from songs a
JOIN play_history b on a.song_id = b.song_id
Where b.played_count>9000;
```

Select * from most_played;

song_name	song_popularity
Everybody Wants To Rule The World	86
Summer Of '69	81
...Baby One More Time	78
4 Minutes (feat. Justin Timberlake & Timbaland)	70
Samjhawan	64
Ilahi	67
Love The Way You Lie	82
Breakaway	64

17) Find a tweet which mentions a song whose artist name is 'eminem'.

```
CREATE VIEW eminem_songs AS
SELECT a.tweet_text,b.song_name from music.tweets a
join music.songs b on a.song_id = b.song_id
where a.tweet_text like '%Eminem%';

select * from eminem_songs;
```

tweet_text	song_name
o tanto que eu ouço essa música até hoje e até...	Love The Way You Lie
At a winery with live music...this group is doing ...	Love The Way You Lie
Eminem - Love The Way You Lie ft. Rihanna htt...	Love The Way You Lie

18) Search for a song from our database and find it in twitter tweets and show the number of re-tweets for that tweet.

```
CREATE VIEW song_retweets AS
select a.tweet_text, b.song_name, b.song_artist, a.retweet
from music.tweets a join music.songs b on a.song_id = b.song_id
where a.tweet_text like '%Sweet%';

select * from song_retweets;
```

tweet_text	song_name	song_artist	retweet
#NowPlaying Eurythmics - Sweet Dreams (Are ...	Sweet Dreams (Are Made of This) - Remastered	Eurythmics, Annie Lennox, Dave Stewart	0
Eurythmics, Annie Lennox, Dave Stewart - Swe...	Sweet Dreams (Are Made of This) - Remastered	Eurythmics, Annie Lennox, Dave Stewart	0
Eurythmics-Sweet Dreams (Are Made of This) (...	Sweet Dreams (Are Made of This) - Remastered	Eurythmics, Annie Lennox, Dave Stewart	1
Eurythmics, Annie Lennox, Dave Stewart - Swe...	Sweet Dreams (Are Made of This) - Remastered	Eurythmics, Annie Lennox, Dave Stewart	0
🎶 Sweet Dreams (Are Made of This) (Remaster...	Sweet Dreams (Are Made of This) - Remastered	Eurythmics, Annie Lennox, Dave Stewart	0

19) Fetch songs and retweets tweeted by spotify twitter handle and find similar results in your music database.

```
CREATE VIEW spotify_twitter_songs AS
select a.tweet_text,b.song_name,b.song_artist,a.retweet
from music.tweets a join music.songs b on a.song_id = b.song_id
where a.twitter_handle like '%AM25spotify%';

select * from spotify_twitter_songs;
```

tweet_text	song_name	song_artist	retweet
Everybody Wants To Rule The World - Tears Fo...	Everybody Wants To Rule The World	Tears For Fears	0

20) List the songs with genre where played count is more than 5000

```
CREATE VIEW 5000played AS
select b.song_name, b.song_genre, c.played_count
from music.songs b Join music.play_history c on b.song_id = c.song_id
where c.played_count > 5000;
```

select * from 5000played;

song_name	song_genre	played_count
Eye of the Tiger	Rock	7647
Everybody Wants To Rule The World	R&B	9455
In Da Club	Hip-Hop	7498
Black and Yellow	Rap	7111
Let Me Love You	Pop	8560
New Rules	Pop	5967
Around the World (La La La La La) - Radio Version	Pop	5263
Not Afraid	Hip-Hop	7407

Source of Data:

Finding the right source of data will help to get appropriate and good quality of data for database. For this project, our source of data is spotify and we have used exportify to export the data of spotify.

CODE:

The below code is the code we have written for scraping the twitter and getting the data and inserting into the tables we had created for twitter data in our database. As mentioned in the comments, we are making database connections, authenticating the user, getting all the tweets for a particular time period, and then insert the tweets or data we retrieved into respective tables in the database. And some queries regarding the data we got from twitter scraping.

```python
import tweepy # for twitter api
import time
import mysql.connector
from mysql.connector import Error
import re
import requests
from requests_oauthlib import OAuth1
import json
import datetime
from datetime import datetime, timedelta

#database connection
def create_connection(host_name, user_name, user_password, db):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password,
            db=db
        )
        print("Connection to MySQL DB successful")
    except Error as e:
        print("The error '{e}' occurred")
    return connection
connection = create_connection("localhost", "root", "admin", "music")
cursor = connection.cursor()

#  Authenticate
APP_KEY = "xZ4OSNBnmkE9zN67GUBhPCqSB"
APP_SECRET = "tb57HaW5EzgGUU5pDC43JkRVxZ9f3YbEUOF7AEWFjuiUIQ7GSU"
USER_OAUTH_TOKEN = "1587541442548838400-yzsSqXr7g5vmAhLq5N3GPQapKhd1TH"
USER_OAUTH_TOKEN_SECRET = "g5XvNK7Lg8wmyg6UCXE2QJ3FgpGrdXjKzcn51KbR9aUe9"
BEARER_TOKEN = "AAAAAAAAAAAAAAAAAAAAABFZiwEAAAAA3mCApctlxrQEXK9%2FiGk6qNpTnXI%3D8ctkG7OyB1Tm7Mj4LSf07I6q5I4qPh1fmt0en0dUF06Z5Kwha1"
authApi = OAuth1(APP_KEY, APP_SECRET,
                 USER_OAUTH_TOKEN, USER_OAUTH_TOKEN_SECRET)

auth = tweepy.OAuthHandler(APP_KEY, APP_SECRET)
api = tweepy.API(auth)

#  Retrieve Tweets
cursor.execute('SELECT * FROM songs s where s.song_popularity > 84')
songs = cursor.fetchall()
```

```python

# Map song name and song id
song_connect=  {}
for song in songs:
    song_connect[song[1]]=song[0]

keywords=[]
for i in songs:
    keywords.append(i[1])
    print(i[1])
print(keywords)

for names in keywords:

    public_tweets = api.search_tweets(names)
    song_id = song_connect[names]
    print("-----------------------------", public_tweets)

    for tweet in public_tweets:
        # Using tweets only from 2022-11-01 to 2022-11-12
        t1=datetime.strptime("2022-11-01","%Y-%m-%d")
        t2=datetime.strptime("2022-11-12","%Y-%m-%d")
        t1date=t1.date()
        t2date=t2.date()
        checkdate= tweet.created_at.date()

        if(checkdate<t1date and checkdate>t2date):
            break
        else:
            tweet_id = tweet.id
            created_at = tweet.created_at
            tweet_text = tweet.text
            username = tweet.user.screen_name
            name = tweet.user.name
            userId = tweet.user.id
            follower_count = tweet.user.followers_count
            following_count = tweet.user.friends_count
            twitter_handle = tweet.user.screen_name
            profile_image_url = tweet.user.profile_image_url_https
            description = tweet.user.description
            userCreated_at = tweet.user.created_at

```

```python

            cursor.execute('''select * from tweets WHERE tweet_id = %s''', (tweet_id, ))
            findTweet = cursor.fetchone()
            print("findTweet----------", findTweet)
            if findTweet:
                print("Tweet already exists")
            else:
                cursor.execute('''select * from twitter_user where twitter_handle = %s''', (twitter_handle, ))
                findUser = cursor.fetchone()
                print("findUser----------", findUser)
                if findUser:
                    print("User already exists")
                else:
                    #insert into twitter_user table
                    cursor.execute('''insert into twitter_user (twitter_handle, name,  description, followers_count, following_count, created_at) values
                    (%s, %s, %s, %s, %s, %s)''', (twitter_handle, name, description, follower_count, following_count, userCreated_at))

                tweets = api.get_status(tweet_id)
                favorite_count = tweets.favorite_count
                retweet_count = tweets.retweet_count

                #insert into tweets table
                cursor.execute('''insert into tweets (tweet_id, twitter_handle, tweet_text, created_at, song_id, likes, retweet) values
                (%s, %s, %s, %s, %s, %s, %s);''', (tweet_id, twitter_handle, tweet_text, created_at, song_id, favorite_count, retweet_count))

                if(len(tweet.entities['user_mentions']) > 0):
                    for mention in tweet.entities['user_mentions']:
                        target_user = mention['screen_name']
                        #insert into tweet_mentions table
                        cursor.execute('''insert into tweet_mentions (tweet_id, source_user, target_user) values (%s, %s, %s)''', (tweet_id, twitter_handle, target_user))

                if(len(tweet.entities['hashtags']) > 0):
                    for tag in tweet.entities['hashtags']:
                        tag = tag['text']
                        #insert into tweet_tags table
                        cursor.execute('''insert into tweet_tags (tweet_id, tags) values (%s, %s)''', (tweet_id, tag))

                connection.commit()
                print(cursor.rowcount, "was inserted.")

```

```python
# Q1    What user posted this tweet?
anyTweet = input("Enter tweet id: ")
first = cursor.execute("SELECT u.name, t.tweet_text FROM twitter_user as u INNER JOIN tweets as t ON u.twitter_handle = t.twitter_handle WHERE t.
first = cursor.fetchone()
print("Q1: ", first)

# Q2    When did the user post this tweet?
second = cursor.execute("SELECT u.name, t.tweet_text, t.created_at FROM twitter_user as u INNER JOIN tweets as t ON u.twitter_handle = t.twitter_
second = cursor.fetchone()
print("Q2: ", second)

# Q3    What tweets have this user posted in the past 24 hours?
anyUser = input("Enter user handle: ")
now = datetime.now()
prev = now + timedelta(days=-1)
now = datetime.strftime(now, "%Y-%m-%d %H:%M:%S")
prev = datetime.strftime(prev, "%Y-%m-%d %H:%M:%S")
print("time------", type(prev), now)
fetchThree = api.user_timeline(screen_name=anyUser, count=100)
for tweet in fetchThree:
    created_at_date = datetime.strftime(tweet.created_at, "%Y-%m-%d %H:%M:%S")
    print(created_at_date)
    if(created_at_date > prev and created_at_date < now):
        print("tweet in 24 hours")
        cursor.execute("insert into tweets (tweet_id, twitter_handle, tweet_text, created_at, likes, retweet) values (%s, %s, %s, %s, %s, %s);",
        (tweet.id, anyUser, tweet.text, tweet.created_at, tweet.favorite_count, tweet.retweet_count))
        connection.commit()
        print(cursor.rowcount, "was inserted.")
    else:
        print("tweet not in 24 hours")
qthree = cursor.execute("SELECT u.name, t.tweet_text, t.created_at FROM twitter_user as u INNER JOIN tweets as t ON u.twitter_handle = t.twitter_
qthree = cursor.fetchall()
print("Q3: ", qthree)
```

```python
# Q4    How many tweets have this user posted in the past 24 hours?
fourth = cursor.execute("SELECT u.name, count(t.tweet_text), t.created_at FROM twitter_user as u INNER JOIN tweets as t ON u.twitter_handle = t.tw
fourth = cursor.fetchall()
print("Q4: ", fourth)

# Q5    When did this user join Twitter?
qfive = cursor.execute("SELECT u.twitter_handle, u.created_at FROM twitter_user as u where u.twitter_handle = %s", (anyUser, ))
qfive = cursor.fetchall()
print("Q5: ", qfive)
# Q6    What keywords/ hashtags are popular?
qsix = cursor.execute("SELECT t.tags, count(t.tags) FROM tweet_tags as t GROUP BY t.tags ORDER BY count(t.tags) DESC LIMIT 10")
qsix = cursor.fetchall()
print("Q6: ", qsix)
# Q7    What tweets are popular?
qseven = cursor.execute("SELECT t.tweet_text, t.likes, t.retweet FROM tweets as t ORDER BY t.retweet DESC LIMIT 10")
qseven = cursor.fetchall()
print("Q7: ", qseven)
```

The scraped data was raw data that we audited, cleaned, and validated with completeness. This involved downloading and reformatting the raw data.

DATABASE PROJECT QUALITY

Audit Validity/ Accuracy, Audit Completeness, Audit Consistency/Uniformity:

A data audit is a step-by-step process that examines every step of the data science process. Problems can be introduced at any step of this process, so a full audit requires close examination at each step. Generally, auditing refers to inspecting an item or a process.

Once data is collected it is important to validate/audit that data to ensure that it can be used for effective decision-making. The intent of Completeness testing is to evaluate whether you have received a full set of data - data anomalies can be explored further after the preliminary testing. This also involved sanity checks. Some of the things considered during this process are below.

```python
#import the libraries required
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
%matplotlib inline
```

Python

```python
#read the csv file
import pandas as pd
data = pd.read_csv("D:\Music-recommendation-database-main\Music-recommendation-database-main\Data\sourceData.csv")
data
```

Python

	Song_id	Track Name	Artist URI(s)	Artist Name(s)	Album URI	Album Name	Album Artist URI(s)	Track URI	Album Artist Name(s)	Album Release Date	
0	1	Beat It	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:1C2h7mLntPSeVYciMRTF4a	Thriller 25 Super Deluxe Edition	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:1OOtq8tRnDM8kG2gqUPjAj	Michael Jackson	11/30/1982	https://i.scdn.co/image/ab6761
1	2	Smooth Criminal - 2012 Remaster	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:24TAupSNVWSAHL0R7n71vm	Bad 25th Anniversary	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:2bCQHF9gdG5BNDVuFlEnNk	Michael Jackson	8/31/1987	https://i.scdn.co/image/ab67616
2	3	Gimme More	spotify:artist:26dSoYclwsYLMAKD3tpOr4	Britney Spears	spotify:album:1ePkYcH5ZQCb1b4tQeiEDj	Blackout	spotify:artist:26dSoYclwsYLMAKD3tpOr4	spotify:track:6ic8OIlUNEATToEFU3xmaH	Britney Spears	10/25/2007	https://i.scdn.co/image/ab6761
3	4	Eye of the Tiger	spotify:artist:26bcq2nyj5GB7uRr558iQg	Survivor	spotify:album:3t3BbpFJGcXI4jI5CRLLA	Rocky IV	spotify:artist:0LyfQWJT6nXafLPZqxe9Of	spotify:track:2KH16WveTQWT6KOG9Rg6e2	Various Artists	6/7/1905	https://i.scdn.co/image/ab6761
4	5	Everybody Wants To Rule The World	spotify:artist:4bthk9UfsYUYdcFyqxmSUU	Tears For Fears	spotify:album:3myPwaMYjdwhtq0nFgeG6W	Songs From The Big Chair (Super Deluxe Edition)	spotify:artist:4bthk9UfsYUYdcFyqxmSUU	spotify:track:4RvWPyQ5RL0ao9LPZeSouE	Tears For Fears	2/25/1985	https://i.scdn.co/image/ab67616
...	...	...	...	...	...	...	...	...	...	...	
315	316	Beat It / State of Shock - Immortal Version	spotify:artist:3fMbdgg4jU18AjLCKBhRSm, spotify...	Michael Jackson, The Jacksons, Mick Jagger, Ke...	spotify:album:5ReKddpdZKgpVlecQhLnEH	Immortal	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:3ejjvP4oU6RoyiK9xZ5lwU	Michael Jackson	11/21/2011	https://i.scdn.co/image/ab67616
316	317	Thriller - Immortal Version	spotify:artist:3fMbdgg4jU18AjLCKBhRSm, spotify...	Michael Jackson, Kevin Antunes	spotify:album:5ReKddpdZKgpVlecQhLnEH	Immortal	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:4V79uSLjxBSe1heEA2vWHO	Michael Jackson	11/21/2011	https://i.scdn.co/image/ab67616
317	318	Smooth Criminal - Immortal Version	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:5ReKddpdZKgpVlecQhLnEH	Immortal	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:4jFI904O3cx2wNMMIXAwlw	Michael Jackson	11/21/2011	https://i.scdn.co/image/ab67616
318	319	Thriller	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:7pMVCMwGykuEu9rzTHxLCm	Michael Jackson's This Is It	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:1D9KEXlrImPUkMTdYzqgX4	Michael Jackson	10/26/2009	https://i.scdn.co/image/ab67616
319	320	Slave to the Rhythm - Audien Remix Radio Edit	spotify:artist:3fMbdgg4jU18AjLCKBhRSm, spotify...	Michael Jackson, Audien	spotify:album:64pX2ZupRbnsRz1cnox0ss	Slave to the Rhythm (Audien Remix Radio Edit)	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:25L1bJsa5pc1q7U9gB40UC	Michael Jackson	8/12/2014	https://i.scdn.co/image/ab67616

320 rows × 18 columns

```python
#sort the data according to no of times played
data = data.sort_values('No of times Songs Played')
data
```

Python

	Song_id	Track Name	Artist URI(s)	Artist Name(s)	Album URI	Album Name	Album Artist URI(s)	Track URI	Album Artist Name(s)	Album Release Date	
261	262	Gravitas	spotify:artist:7gzzt6L2XRlxnJsIL0Ytve	Advakit	spotify:album:7y1Yfj5tKzKQ5TLEZvuYvY	Gravitas	spotify:artist:7gzzt6L2XRlxnJsIL0Ytve	spotify:track:2H4vzvm58tJwRmDF5IessU	Advakit	5/1/2020	https://i.scdn.co
273	274	Aesthetic Arrest - Live	spotify:artist:5XMQ9F7g4nCx2VCtrnPm7d	Evanoff	spotify:album:3aIqc6KE5bVJefs2hw3wFN	Evanoff Live	spotify:artist:5XMQ9F7g4nCx2VCtrnPm7d	spotify:track:5aC7n8UxU5nX2Njc2E6p6Y	Evanoff	10/11/2018	https://i.scdn.co
282	283	On My Way	spotify:artist:38aAEKHAmZwECZgFuUfdCB, spotify...	Teddy Beats, Mon Rovía	spotify:album:55cqS2XvKGw0RYnUVxAtPF	On My Way	spotify:artist:38aAEKHAmZwECZgFuUfdCB, spotify...	spotify:track:2kqAJgHqgO0VWEvDmvg0ct	Teddy Beats, Mon Rovía	4/10/2020	https://i.scdn.co
37	38	Man in the Mirror - 2012 Remaster	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:24TAupSNVWSAHL0R7n71vm	Bad 25th Anniversary	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:1kiNatlrwDusOZfR29W0LJ	Michael Jackson	8/31/1987	https://i.scdn.co/
292	293	Raveolution	spotify:artist:53UXMZxwzQyV4j7tZaVF58, spotify...	Sandro Silva, Graham Bell	spotify:album:6RpNpBfHqKAwAfM8kaK4KA	Raveolution	spotify:artist:53UXMZxwzQyV4j7tZaVF58, spotify...	spotify:track:3zi577X0DUUbQKUcLuTHNb	Sandro Silva, Graham Bell	3/16/2020	https://i.scdn.co
...	...	...	...	...	...	...	...	...	...	...	
97	98	(Your Love Keeps Lifting Me) Higher & Higher	spotify:artist:4VnomLtKTm9Ahe1tZfmZju	Jackie Wilson	spotify:album:4OgYmImCWz3VK7At9GTWHN	Higher And Higher	spotify:artist:4VnomLtKTm9Ahe1tZfmZju	spotify:track:5qyq1H5OPMlfuvZQ1wQNo7	Jackie Wilson	1967	https://i.scdn.co/
128	129	The Other Way	spotify:artist:4YepO0c4kXzTyaRgzvhhTb, spotify...	Bastiqe, Encure	spotify:album:5cvleL27NYmwQnWc3Zeymz	The Other Way	spotify:artist:4YepO0c4kXzTyaRgzvhhTb, spotify...	spotify:track:6AAYNPXVE6SCFMJsrbf7db	Bastiqe, Encure	6/4/2021	https://i.scdn.co
318	319	Thriller	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:7pMVCMwGykuEu9rzTHxLCm	Michael Jackson's This Is It	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	spotify:track:1D9KEXlrImPUkMTdYzqgX4	Michael Jackson	10/26/2009	https://i.scdn.co/
232	233	Don't Let Me Down	spotify:artist:69GGBxA162lTqCwzJG5jLp, spotify...	The Chainsmokers, Daya	spotify:album:2SByipSK8eZ2pasalwwzhf	Don't Let Me Down	spotify:artist:69GGBxA162lTqCwzJG5jLp	spotify:track:1i1fxkWeaMmKEB4T7zqbzK	The Chainsmokers	2/5/2016	https://i.scdn.co
223	224	Inception	spotify:artist:586NPHDKjGokvJNJJeRyoS	Aleton	spotify:album:2HYoi9uL9Jb0dIRu2xKbsg	Inception	spotify:artist:586NPHDKjGokvJNJJeRyoS	spotify:track:79Hg6pgXLHeffOTOIafD3d	Aleton	7/3/2020	https://i.scdn.co

320 rows × 18 columns

```python
for column in data:
    print(data[column].values)
```

[5] Python

```python
corelation = data.corr()
```

[6] Python

```
C:\Users\Abhishek\AppData\Local\Temp\ipykernel_25168\1518246052.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a
future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corelation = data.corr()
```

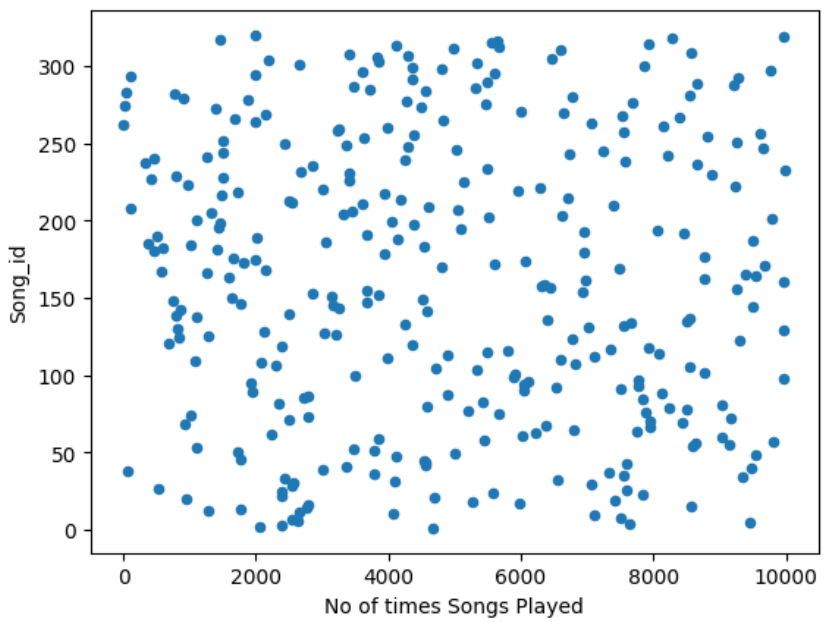
```python
print(corelation.columns)
```

[7] Python

```
Index(['Song_id', 'Disc Number', 'Track Number', 'Track Duration (ms)',
       'Popularity', 'No of times Songs Played'],
      dtype='object')
```

```python
data.plot(kind='scatter', y = 'Song_id', x='No of times Songs Played');
```

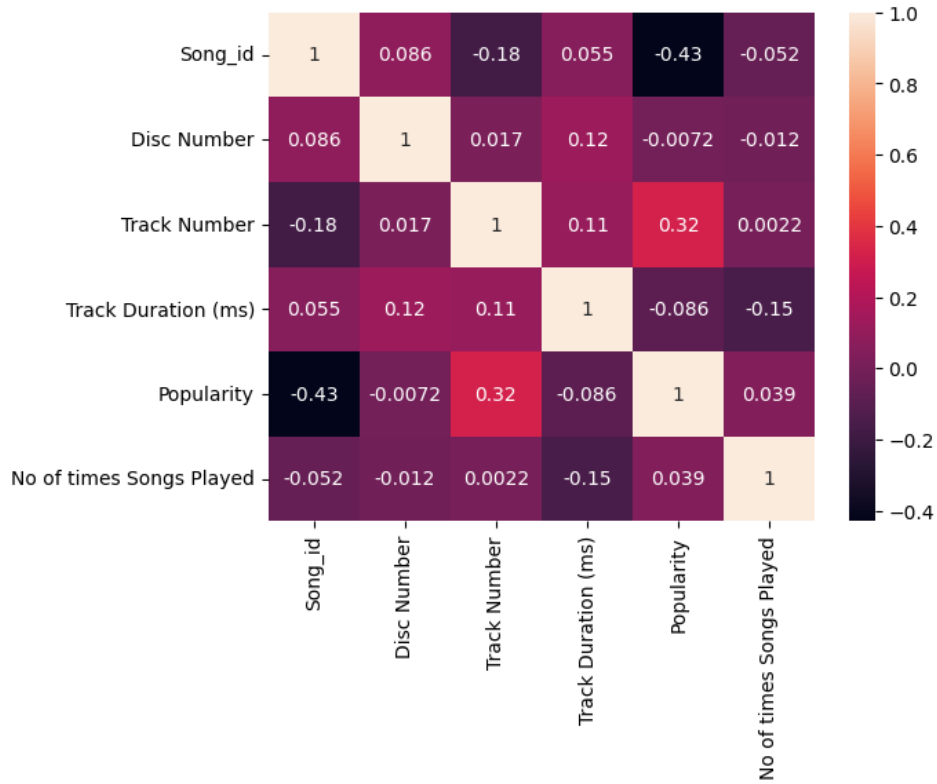
[8] Python




```python
sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True)
```

[9] Python

```
<AxesSubplot: >
```




```python
# Audit completeness
data2 = pd.read_csv("D:\Music-recommendation-database-main\Music-recommendation-database-main\Data\sourceData.csv")
data2['Track Name'].describe()
```

[22] Python

```
count                                          316
unique                                         293
top       Around the World (La La La La La) - Radio Version
freq                                             4
Name: Track Name, dtype: object
```

```python
data2.isnull().sum()
```

[11] Python

```
Song_id                   0
Track Name                4
Artist URI(s)             5
Artist Name(s)            3
Album URI                 7
Album Name                4
Album Artist URI(s)       6
Track URI                 5
Album Artist Name(s)      4
Album Release Date        4
Album Image URL           6
Disc Number               5
Track Number              1
Track Duration (ms)       2
Track Preview URL        78
Explicit                  3
Popularity               25
No of times Songs Played  1
dtype: int64
```

```python
data3 = pd.read_csv("D:\Music-recommendation-database-main\Music-recommendation-database-main\Data\cleaned_data.csv")
data3
```

[23] Python

	Song_Id	Song_Name	Artist_Name	Album_Name	Album_Artist_Name	Release_Year	Duration	Explicit	Popularity	Count_of_Plays
0	1	Beat It	Michael Jackson	Thriller 25 Super Deluxe Edition	Michael Jackson	1982	4:18	False	81	4680
1	2	Smooth Criminal - 2012 Remaster	Michael Jackson	Bad 25th Anniversary	Michael Jackson	1987	4:18	False	79	2055
2	3	Gimme More	Britney Spears	Blackout	Britney Spears	2007	4:11	False	78	2397
3	4	Eye of the Tiger	Survivor	Rocky IV	Various Artists	1905	4:06	False	76	7647
4	5	Everybody Wants To Rule The World	Tears For Fears	Songs From The Big Chair (Super Deluxe Edition)	Tears For Fears	1985	4:11	False	86	9455
...	...	...	...	...	...	...	...	...	...	...
290	316	Beat It / State of Shock - Immortal Version	Michael Jackson, The Jacksons, Mick Jagger, Ke...	Immortal	Michael Jackson	2011	3:09	False	37	5641
291	317	Thriller - Immortal Version	Michael Jackson, Kevin Antunes	Immortal	Michael Jackson	2011	3:38	False	42	1461
292	318	Smooth Criminal - Immortal Version	Michael Jackson	Immortal	Michael Jackson	2011	2:00	False	56	8275
293	319	Thriller	Michael Jackson	Michael Jackson's This Is It	Michael Jackson	2009	5:57	False	60	9973
294	320	Slave to the Rhythm - Audien Remix Radio Edit	Michael Jackson, Audien	Slave to the Rhythm (Audien Remix Radio Edit)	Michael Jackson	2014	3:14	False	32	1983

295 rows × 10 columns

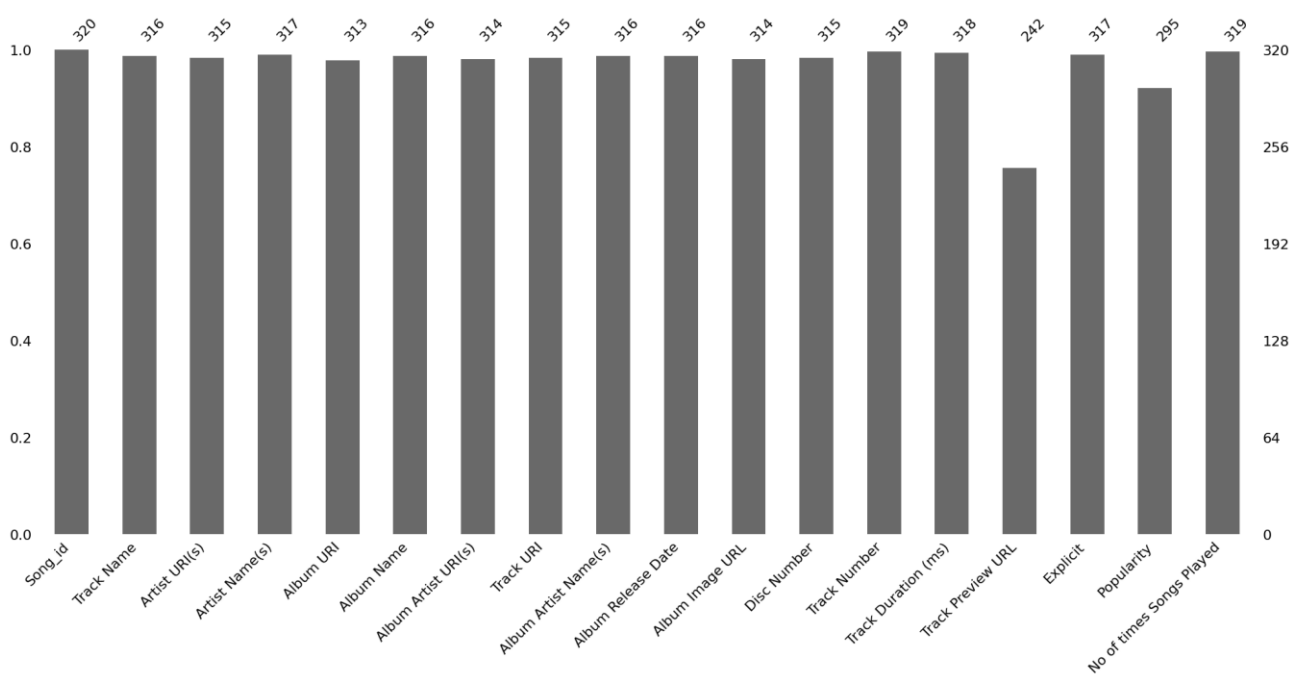
```python
#check for null data
data3.isnull().sum()
```

[13] Python

```
Song_Id             0
Song_Name           0
Artist_Name         0
Album_Name          0
Album_Artist_Name   0
Release_Year        0
Duration            0
Explicit            0
Popularity          0
Count_of_Plays      0
dtype: int64
```

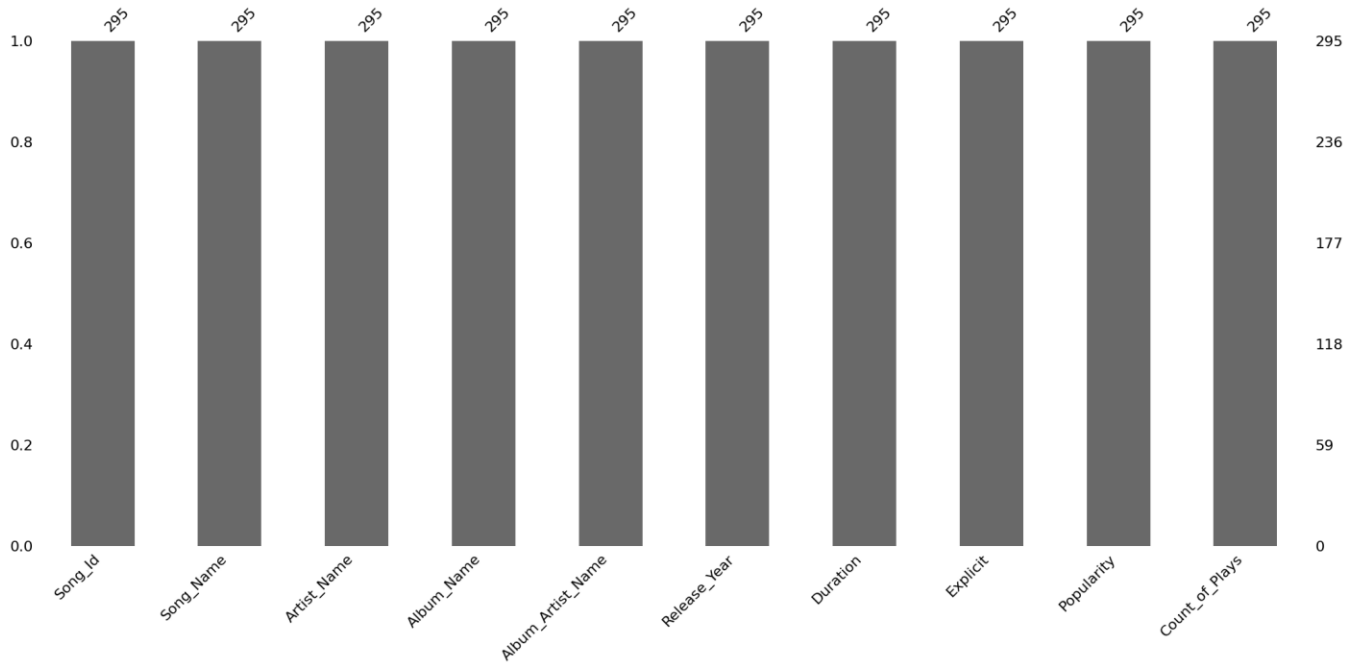
```python
#plot a bar graph to represent null data from source data
msno.bar(data2)
```

[14] Python



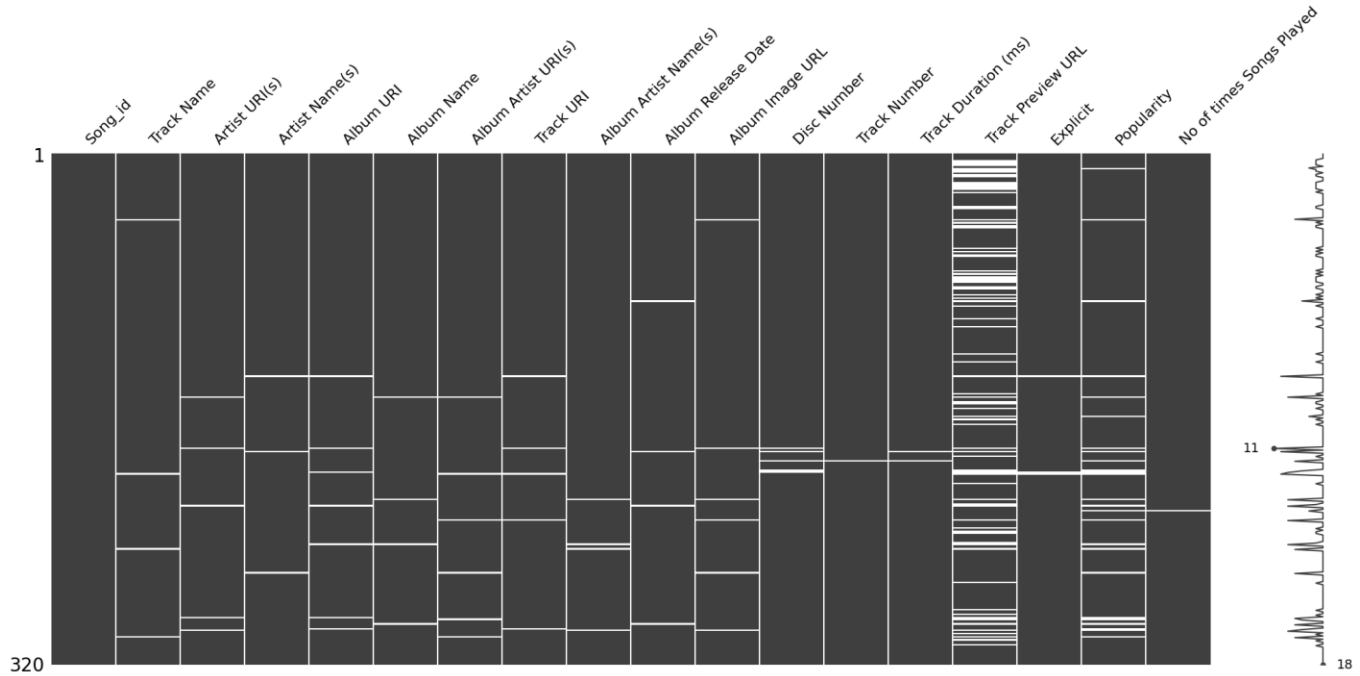
```python
#plot a bar graph to show null data after cleaning
msno.bar(data3)
```

[15] Python




```python
#plot a matrix graph to show null data from source data
msno.matrix(data2)
```

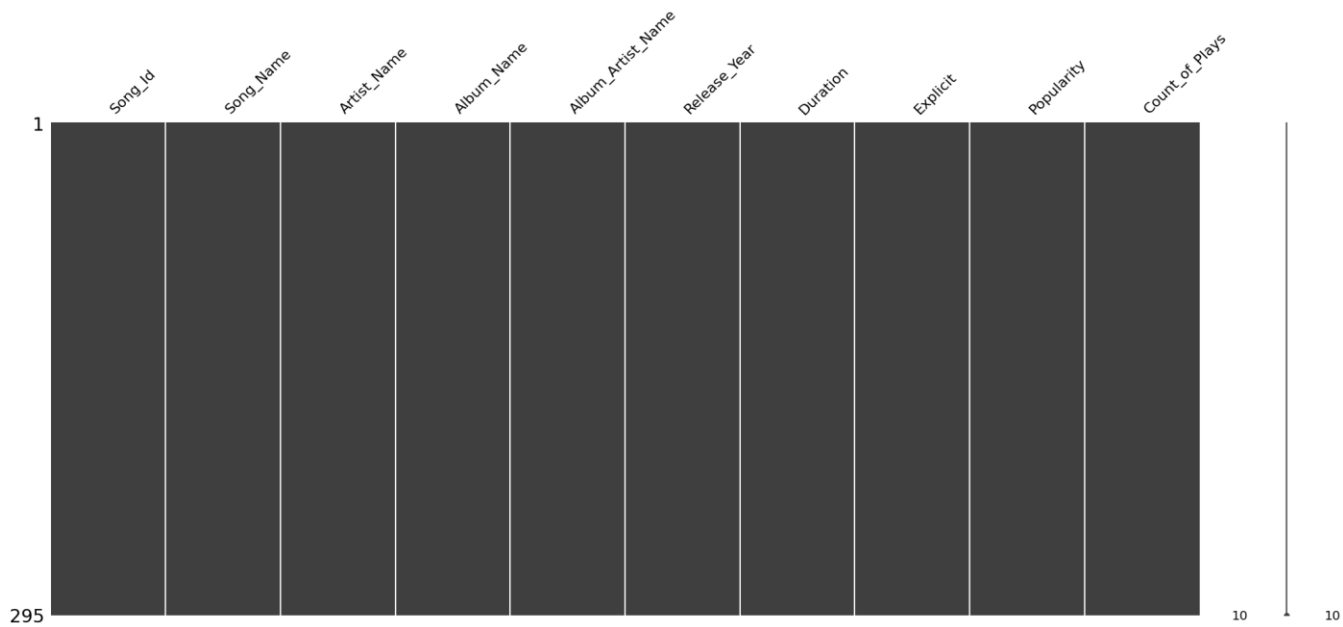
[16] Python



```python
#plot a matrix graph to show null data after cleaning of data
msno.matrix(data3)
```

[17] Python

```
<AxesSubplot: >
```




```python
#Audit consistency
data4 = pd.read_csv("D:\Music-recommendation-database-main\Music-recommendation-database-main\Data\Consistency.csv")
data4
```

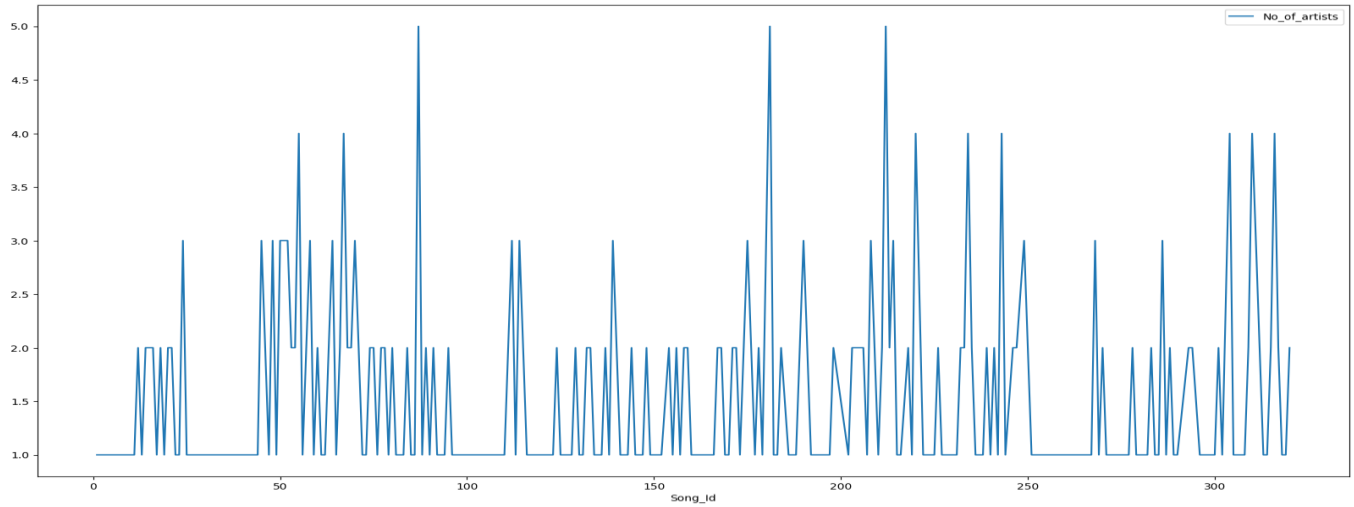
[24] Python

	Song_Id	Song_Name	No_of_artists
0	1	Beat It	1
1	2	Smooth Criminal - 2012 Remaster	1
2	3	Gimme More	1
3	4	Eye of the Tiger	1
4	5	Everybody Wants To Rule The World	1
...	...	...	...
290	316	Beat It / State of Shock - Immortal Version	4
291	317	Thriller - Immortal Version	2
292	318	Smooth Criminal - Immortal Version	1
293	319	Thriller	1
294	320	Slave to the Rhythm - Audien Remix Radio Edit	2

295 rows × 3 columns

```python
data4.plot(x="Song_Id", y="No_of_artists", kind="line", figsize=(20, 10))
plt.show()
#Each record of song has at least one artist
```

[28] Python



DATA CLENSING PROCESS BEFORE THE FORMATION OF FINAL DATABASE:

- After analyzing the data in the dataset exported from the Spotify platform, relevant columns and data were retained as per the table structure in the database.
- After the structural cleansing, the data was fixed at an individual level by adding constant values, to avoid complete data loss.
- Hence, Maximum Data was retained from the imported dataset.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import mysql.connector
from mysql.connector import Error
import random
import os
import os.path

d = os.getcwd()
# os.chdir("..")
o = [os.path.join(d,o) for o in os.listdir(d) if os.path.isdir(os.path.join(d,o))] # Gets all directories in the folder as a tuple

#database connection
def create_connection(host_name, user_name, user_password, db):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password,
            db=db
        )
        print("Connection to MySQL DB successful")
    except Error as e:
        print("The error '{e}' occurred")
    return connection
connection = create_connection("localhost", "root", "admin", "music")
cursor = connection.cursor()
```

```python

df = pd.read_csv("C:\\Users\\Abhishek\\Downloads\\new Source data.csv",encoding='latin-1')

#Check for null values and remove them
# df.describe()
print(df.isnull().sum())
print(df.describe)
#remove records with null values
modifiedData = df.dropna()
print(modifiedData.isnull().sum())
print(modifiedData.describe)
modifiedData.to_csv("C:\\Users\\Abhishek\\Downloads\\dataAfterCleaning.csv",index=False)

```

Pre-Clean Dataset Snapshot:

	A	B	C	D	E
1	Song_id	Track Name	Artist URI(s)	Artist Name(s)	Album URI
2	1	Beat It	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:1C2h7mLntPSeVYciMRTF4a
3	2	Smooth Criminal - 2012 Remaster	spotify:artist:3fMbdgg4jU18AjLCKBhRSm	Michael Jackson	spotify:album:24TAupSNVWSAHL0R7n71v
4	3	Gimme More	spotify:artist:26dSoYclwsYLMAKD3tpOr4	Britney Spears	spotify:album:1ePkYcH5ZQCb1b4tQeiEDj
5	4	Eye of the Tiger	spotify:artist:26bcq2nyj5GB7uRr558iQg	Survivor	spotify:album:3t3BbpFJiGcXl4jI5CRLLA
6	5	Everybody Wants To Rule The Worl	spotify:artist:4bthk9UfsYUYdcFyqxmSUU	Tears For Fears	spotify:album:3myPwaMYjdwhtq0nFgeG6
7	6	Everybody Talks	spotify:artist:0RpddSzUHfncUWNJXKOsjy	Neon Trees	spotify:album:0uRFz92JmjwDbZbB7hEBIr
8	7	Earned It (Fifty Shades Of Grey)	spotify:artist:1Xyo4u8uXC1ZmMpatF05PJ	The Weeknd	spotify:album:0P3oVJBFOv3TDXlYRhGL7s
9	8	In Da Club	spotify:artist:3q7HBObVc0L8jNeTe5Gofh	50 Cent	spotify:album:5G5rgQHzdQnw32SI0WjIo5
10	9	Black and Yellow	spotify:artist:137W8MRPWKqSmrBGDBFSop	Wiz Khalifa	spotify:album:6ZOXiVL8rmk2ATHJiFJhiD
11	10	Superman	spotify:artist:7dGJo4pcD2V6oG8kP0tJRR, spo	Eminem, Dina Ra	spotify:album:1ftvBBcu7jYIvXyt3JWB8S
12	11	In Da Club	spotify:artist:3q7HBObVc0L8jNeTe5Gofh	50 Cent	spotify:album:4ycNE7y1rp5215g1kkqk1P
13	12	Candy Shop	spotify:artist:3q7HBObVc0L8jNeTe5Gofh, sp	50 Cent, Olivia	spotify:album:2pidzXTaHV4WaIJYRxKDCH
14	13	All We Have Is This Moment	spotify:artist:6rclmdstz3sdi2HTLzXJwg	C. SHIROCK	spotify:album:62oOarigBpkjNsQktNUdTY
15	14	rockstar (feat. 21 Savage)	spotify:artist:246dkjvS1zLTtiykXe5h60, spotif	Post Malone, 21	spotify:album:6trNtQUgC8cgbWcqoMYkC
16	15	Let Me Love You	spotify:artist:540vIaP2JwjQb9dm3aArA4, spo	DJ Snake, Justin B	spotify:album:19zH1vHqzkxEsUvN7GaaH0
17	16	Blame (feat. John Newman)	spotify:artist:7CajNmpbOovFoOoasH2HaY, s	Calvin Harris, Joh	spotify:album:48zisMeiXniWLzOQghbPqS
18	17	New Rules	spotify:artist:6M2wZ9GZgrQXHCFfjv46we	Dua Lipa	spotify:album:01sfgrNbnnPUEyz6GZYlt9
19	18	Around the World (La La La La La) -	spotify:artist:5wTdspmxzb8V4ZjvDodpBo, sp	A Touch Of Class	spotify:album:2kBFECL9a71fNRXbRW5xO
20	19	Not Afraid	spotify:artist:7dGJo4pcD2V6oG8kP0tJRR	Eminem	spotify:album:47BiFcV59TQi2s9SkBo2pb
21	20	Love The Way You Lie	spotify:artist:7dGJo4pcD2V6oG8kP0tJRR, spo	Eminem, Rihanna	spotify:album:47BiFcV59TQi2s9SkBo2pb

Post-Clean Dataset Snapshot:

	A	B	C	D	E	F	G	H
1	song_id	song_name	song_album	song_artist	song_genre	song_length	song_language	song_popularity
2	1	Beat It	Thriller 25 Super Deluxe Edition	Michael Jackson	Pop	3	English	81
3	2	Smooth Criminal - 2012 Remaster	Bad 25th Anniversary	Michael Jackson	Pop	3	English	79
4	3	Gimme More	Blackout	Britney Spears	Pop	3	English	78
5	4	Eye of the Tiger	Rocky IV	Survivor	Rock	3	English	76
6	5	Everybody Wants To Rule The World	Songs From The Big Chair (Super D	Tears For Fears	R&B	3	English	86
7	6	Everybody Talks	Picture Show	Neon Trees	R&B	2	English	79
8	7	Earned It (Fifty Shades Of Grey)	Beauty Behind The Madness	The Weeknd	Pop	3	English	74
9	8	In Da Club	Get Rich Or Die Tryin'	50 Cent	Hip-Hop	2	English	82
10	9	Black and Yellow	Rolling Papers	Wiz Khalifa	Rap	3	English	75
11	10	Superman	The Eminem Show	Eminem, Dina Rae	Hip-Hop	4	English	0
12	11	In Da Club	Get Rich Or Die Tryin'	50 Cent	Hip-Hop	2	English	75
13	12	Candy Shop	The Massacre	50 Cent, Olivia	Hip-Hop	2	English	80
14	13	All We Have Is This Moment	All We Have Is This Moment	C. SHIROCK	R&B	3	English	9
15	14	rockstar (feat. 21 Savage)	beerbongs & bentleys	Post Malone, 21 Savage	R&B	3	English	84
16	15	Let Me Love You	Encore	DJ Snake, Justin Bieber	Pop	2	English	49
17	16	Blame (feat. John Newman)	Motion	Calvin Harris, John Newman	Electronic	2	English	76
18	17	New Rules	Dua Lipa (Deluxe)	Dua Lipa	Pop	2	English	80
19	18	Around the World (La La La La La) - Ra	Planet Pop	A Touch Of Class, Pete Konemann	Pop	2	English	67
20	19	Not Afraid	Recovery	Eminem	Hip-Hop	3	English	80
21	20	Love The Way You Lie	Recovery	Eminem, Rihanna	Hip-Hop	3	English	82

NORMALIZATION

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

The main goal of database normalization is to restructure the logical model of a database to:

> Eliminate redundancy.
> Organize data efficiently.
> Reduce the potential for data anomalies (data anomalies are inconsistency in data stored as a result of database operation)

1st Normalization form:

> Each table has a primary key.
> The values in column of a table are atomic (no multi-value attributes).
> There are no repeating groups.

How we converted our database into 1st NF:

Step 1: Recognizing and eliminating the repeating groups in tables.

Step 2: Identifying the primary key for each table.

Step 3: Identifying all the dependencies in each table.

Below is songs table from our music database.

song_id	song_name	song_album	song_artist	song_genre	song_length	song_language	song_popularity
1	Beat It	Thriller 25 Super Deluxe Edition	Michael Jackson	Pop	3	English	81
2	Smooth Criminal	Bad 25th Anniversary	Michael Jackson	Pop	3	English	79
3	Gimme More	Blackout	Britney Spears	Pop	3	English	78
4	Eye of the Tiger	Rocky IV	Survivor	Rock	3	English	76
5	Everybody Wants To Rule The World	Songs From The Big Chair (Super Deluxe Edition)	Tears For Fears	R&B	3	English	86

6	Everybody Talks	Picture Show	Neon Trees	R&B	2	English	79
7	Earned It (Fifty Shades Of Grey)	Beauty Behind The Madness	The Weeknd	Pop	3	English	74
8	In Da Club	Get Rich Or Die Tryin'	50 Cent	Hip-Hop	2	English	82
9	Black and Yellow	Rolling Papers	Wiz Khalifa	Rap	3	English	75
10	Superman	The Eminem Show	Eminem, Dina Rae	Hip-Hop	4	English	0

All the records in the table are atomic and there are no repeating groups.

2nd Normalization Form:

> All requirements from 1st NF must be met.
> Redundant data across multiple rows of table must be moved to a separate table.
> The resulting table must be related to each other by use of foreign key.

Table is in 2nd Normal form when:

- It is in 1 Normal Form, and
- It includes no partial dependencies, i.e., no attribute is dependent on only a portion of primary key. In other words, every non-primary-key attribute is fully functionally dependent on the primary key.

How we converted our database into 2 NF:

After 1 NF,

Step 1: Writing each key component on a separate line.

Step 2: Assigning corresponding dependent attributes.

Below is the artist table from our music database

id	**name**	**age**	**country**	**recordLabel**
1	Michael Jackson	50	USA	Sony
2	Britney Spears	30	USA	Warner
4	Tears For Fears	34	Ireland	CBS
5	Neon Trees	23	Germany	Universal
6	The Weeknd	35	USA	BMG
7	50 Cent	36	Mexico	ARMIND

8	Wiz Khalifa	32	USA	Sony
9	Eminem	33	USA	Warner

For example, recordLable is dependent is fully functionally dependent on the primary key which is artist_id. And all the records in artist table are atomic and no repeating groups.

3rd Normalization Form:

> All requirements of 2 NF must be met.
> Eliminate the keys that do not depend on primary key. i.e., any key that is dependent not only on primary key but also on another field must be moved to another table.

Table is in 3rd Normal Form when following are true:

- It is in 2nd Normal Form.
- It contains no transitive dependencies.
- There are no fields or attributes that does not depend on the key, i.e., non-key attributes must be dependent on key(s) but and only on the key(s).

How we converted our database into 3rd Normal Form:

Step 1: Identified each new determinant (determinant is any attribute whose value determines other values within a row.

Step 2: Identifying attributes dependent on each determinant identified in step 1 and identify dependency.

Step 3: Removing dependent attributes from transitive dependency.

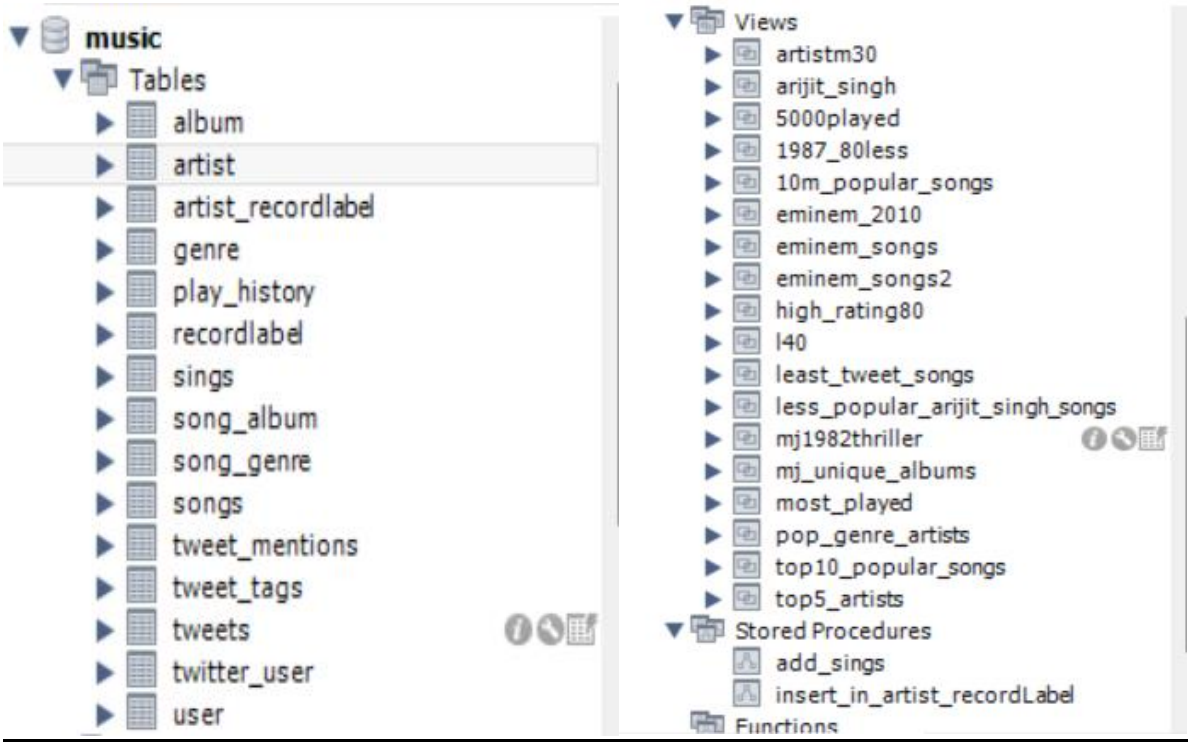
Below is the view from sings table.

song_name	**artist_name**
Beat It	Michael Jackson
Smooth Criminal - 2012 Remaster	Michael Jackson
Gimme More	Britney Spears
Everybody Wants To Rule The World	Tears For Fears
Everybody Talks	Neon Trees
Earned It (Fifty Shades Of Grey)	The Weeknd
In Da Club	50 Cent
Black and Yellow	Wiz Khalifa
In Da Club	50 Cent

'sings' table contains only songs_id and artist_id as attributes which are foreign keys from artist and songs table respectively, using those id's we have fetched the names of songs and artists accordingly.

This shows that there is no partial as well as transitive dependencies present in either songs or artist table.

DATABASE SCHEMA



Below are some examples of SQL Create and insert statements:

Songs Table:

```
CREATE TABLE `songs` (
  `song_id` int NOT NULL AUTO_INCREMENT,
  `song_name` text,
  `song_album` text,
  `song_artist` text,
  `song_genre` text,
  `song_length` bigint DEFAULT NULL,
  `song_language` text,
  `song_popularity` int DEFAULT NULL,
  PRIMARY KEY (`song_id`)
);
```

```
INSERT INTO `music`.`songs`
(`song_id`,`song_name`,`song_album`,`song_artist`,`song_genre`,`song_length`,`song_language`,`song_
popularity`)
VALUES(<{song_id: }>,<{song_name: }>,<{song_album: }>,<{song_artist: }>,<{song_genre:
}>,<{song_length: }>,<{song_language: }>,<{song_popularity: }>);
```

Eg:

```
Insert into Songs Values('In da club','Get rich or die trying','50 cent','Hip-hop',3:13,'English','85');
```

<u>Artist Table:</u>

```
CREATE TABLE `artist` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(25) DEFAULT NULL,
  `age` int DEFAULT NULL,
  `country` varchar(20) DEFAULT NULL,
  `recordLabel` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

```
INSERT INTO `music`.`artist`
(`id`,`name`,`age`,`country`,`recordLabel`)
VALUES
(<{id: }>,<{name: }>,<{age: }>,<{country: }>,<{recordLabel: }>);
```

```
-- insert into sings values
insert into sings(
select a.song_id,b.id from songs a join artist b on a.song_artist = b.name);

-- insert into song_album values
insert into song_album(
select a.song_id,b.id from songs a join album b on a.song_album = b.name);

-- insert into song_genre values
insert into song_genre(
select a.song_id,b.id from songs a join genre b on a.song_genre = b.name);

-- insert into artist)recordlabel values
insert into artist_recordlabel(
select a.id,b.id from artist a join recordlabel b on a.recordLabel = b.name);
```