



NIRVANA

A SONG FOR EVERY MOOD

PROFESSOR SHRUTI VARMA

BY

LEENA BALAGALU

SAI SINDURI VANGALA

TABLE OF CONTENTS

PROJECT SUMMARY.....	3
PURPOSE OF PROJECT.....	4
ER SPREADSHEET	5
CONCEPTUAL DATA MODEL.....	7
LOGICAL DATA MODEL.....	8
BASIC LAYOUT OF POWER APPS.....	9
SQL UP/DOWN SCRIPTS.....	13
TIME LOG OF TEAMMATES.....	27
PRESENTATION SLIDES WITH DEMO OF UI.....	29

PROJECT SUMMARY

The project aims to develop a robust and user-friendly online music streaming platform with features for music search, playlist management, user authentication, administrative control, and basic analytics. Users can register, log in, and manage their profiles, while administrators can curate content and manage the music catalog. The platform operates with a limited music catalog, without real-time streaming or financial transactions. Users can search for music, create playlists. The analytics feature identifies top-rated artists and songs. The project focuses on fundamental functionality, serving as a foundation for potential future enhancements, including more complex features and audio streaming capabilities.

PURPOSE OF PROJECT

The purpose of the project is to create a music platform that's easy to use and enjoyable for people who love music. The focus is on making the experience enjoyable and user-friendly, with features like searching for music, managing playlists, and controlling who has access. The project is not just about what users need now, it's also thinking about how to make things even better in the future. At its core, the project reflects a commitment to the seamless integration of technology and music appreciation. In the long run, the project aspires to be more than a simple platform and aims to be an adaptable and vibrant part of the evolving digital music landscape. The main goal is to create a pleasant and flexible experience that contributes to the idea of how digital platforms can strengthen our connection with music.

ER MODEL

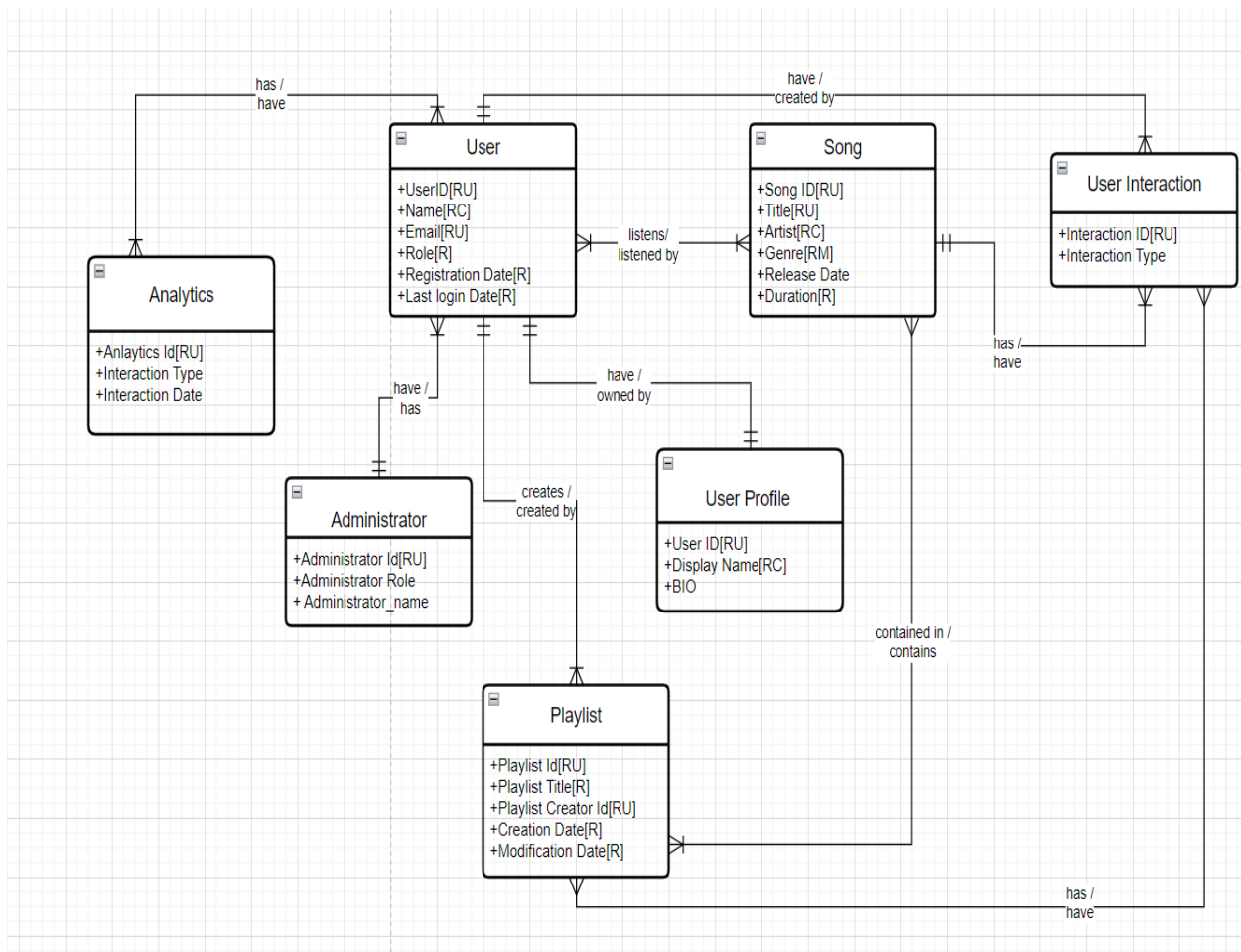
Entities and attributes

Entities and Attributes		
Entity	Attribute	
User	User ID	RU
	User Name	RC
	User Email	RU
	Password	R
	Role	R
	Registration Date	R
	Last login Date	R
Song	Song ID	RU
	Song Title	RU
	Artist	RC
	Genre	RM
	Release Date	
	Song Duration	R
Playlist	Playlist Id	RU
	Playlist Title	R
	Playlist Creator Id	RU
	Creation Date	R
	Modification Date	R
User Interaction	Interaction ID	RU
	Interaction Type	
User Profile	User Profile ID	RU
	Display Name	RC
	BIO	
Analytics	Analytics Id	RU
	Interaction Type	
	Interaction Date	
Administrator	Administrator Id	RU
	Administrator Role	
	Administrator Name	

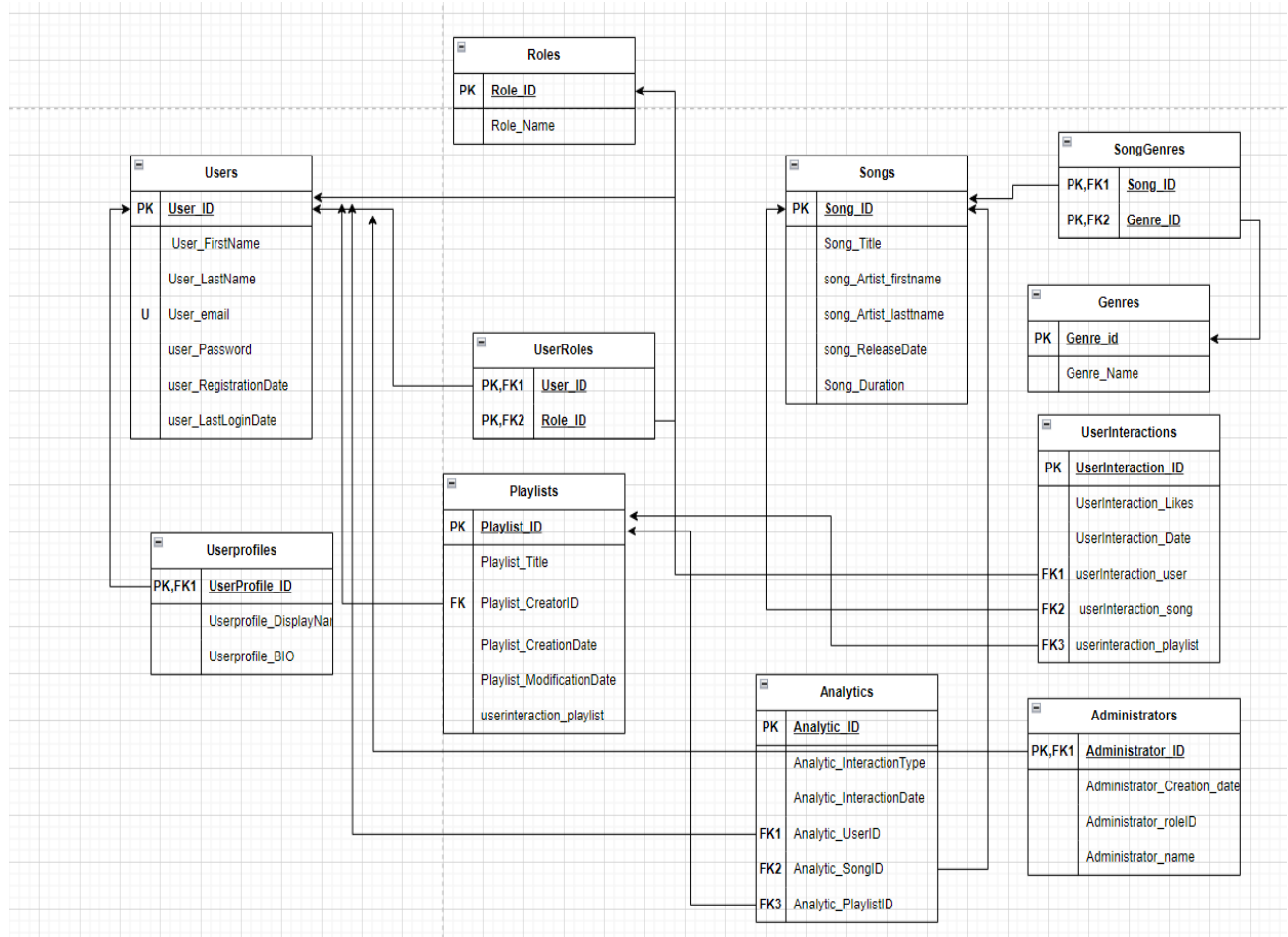
Relationships of entities

Relationship					
Relationship	Entity	Rule	Cardinality		Entity
User,song	user	listens	1	M	song
	song	listened by	1	M	user
user,playlist	user	creates	1	M	playlist
	playlist	created by	1	1	user
user,user interaction	user	have	1	M	user interaction
	user interaction	created by	1	1	user
user,user profile	user	have	1	1	user profile
	user profile	owned by	1	1	user
user,analytics	user	has	1	M	Analytics
	Analytics	have	M	1	user
user,administrator	user	have	1	1	Administrator
	administrator	has	1	M	user
song,playlist	song	contained in	M	M	playlist
	playlist	contains	1	M	songs
song,user interaction	song	has	1	M	user interaction
	user interaction	have	1	1	song
playlist, user interaction	playlist	has	M	M	user interaction
	user interaction	have	M	M	playlist

CONCEPTUAL MODEL

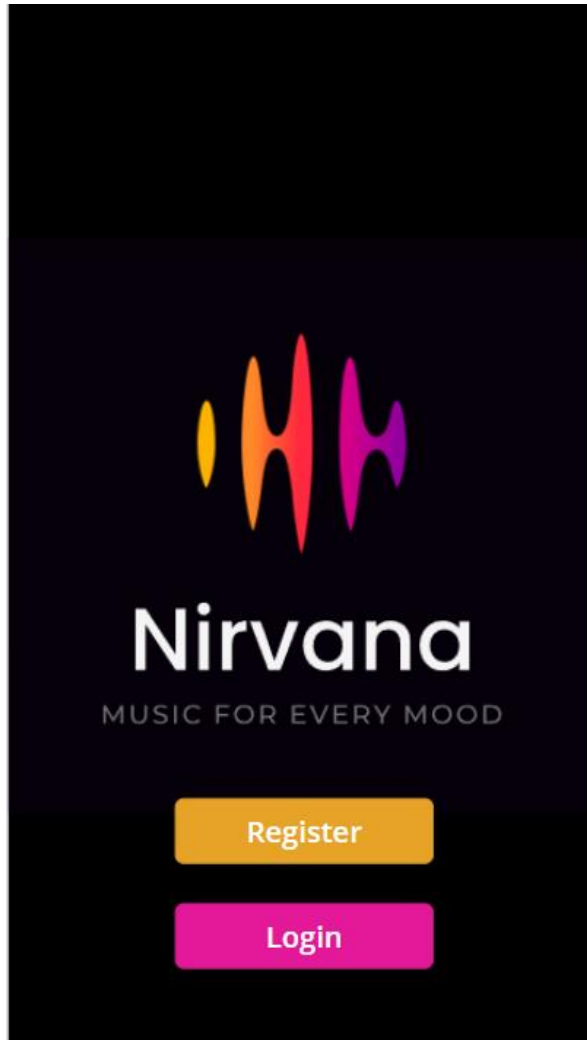


LOGICAL MODEL

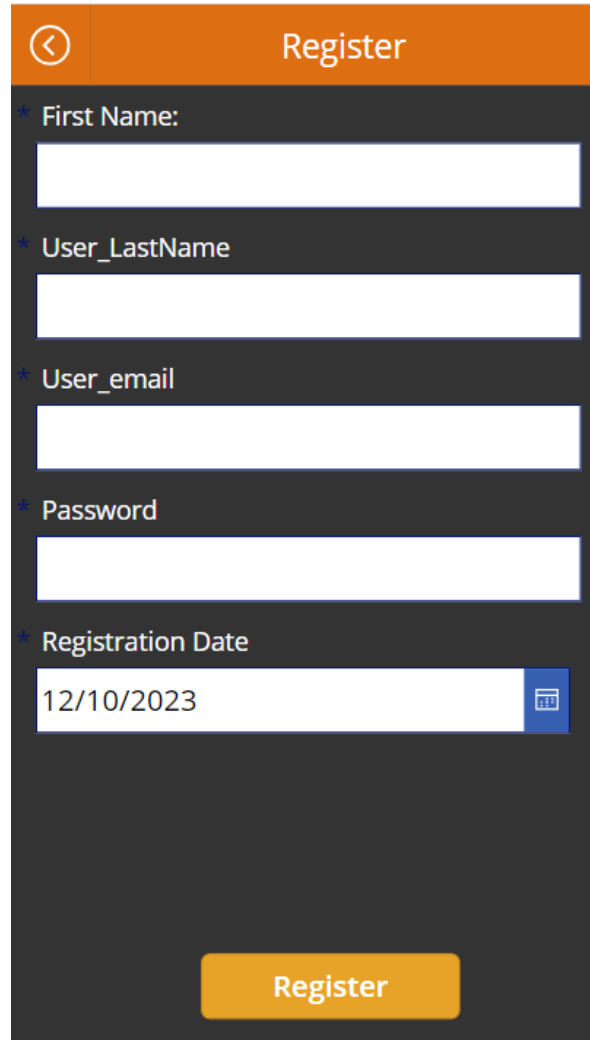


LAYOUT OF POWER APP SCREENS

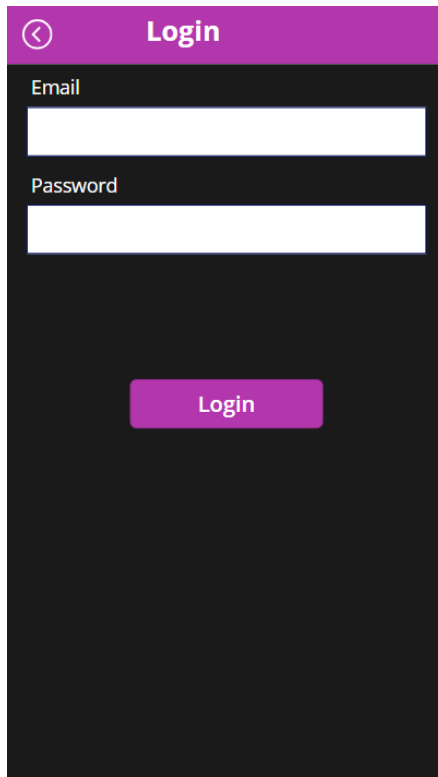
HOME



REGISTER

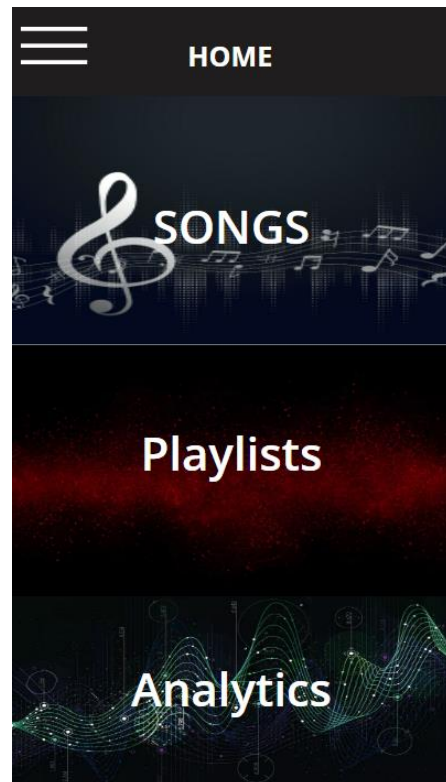
The register screen of the Nirvana app has a dark background. At the top, there is an orange header bar with a white back arrow icon on the left and the word "Register" in white text on the right. Below the header, there are five form fields, each preceded by an asterisk: "First Name:" with a white text input field; "User_LastName" with a white text input field; "User_email" with a white text input field; "Password" with a white text input field; and "Registration Date" with a white text input field containing the date "12/10/2023" and a blue calendar icon on the right. At the bottom of the screen, there is a yellow button with the text "Register".

LOGIN

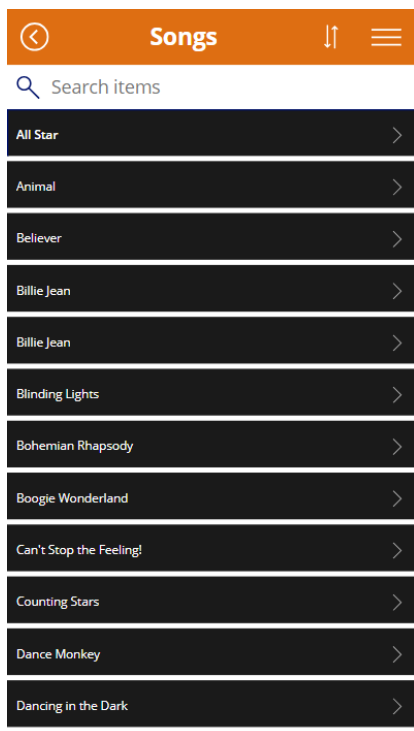


A mobile app login screen with a purple header bar containing a back arrow and the word "Login". Below the header are two white input fields labeled "Email" and "Password". At the bottom is a purple "Login" button.

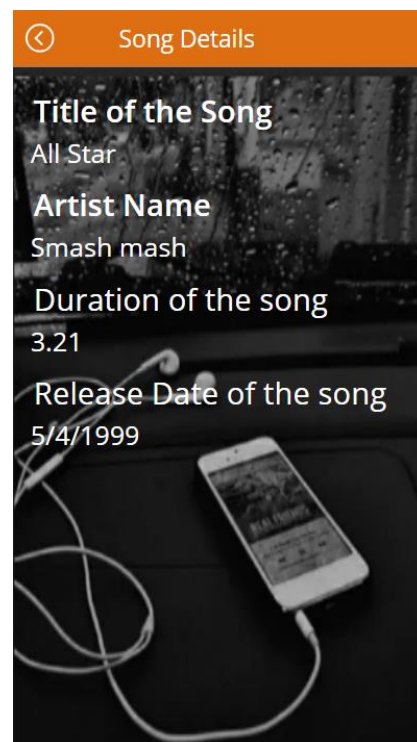
USER HOME





SONGS




SONG DETAILS



PLAYLISTS

 Playlists 

 Search items

Salsa Sensation
Playlist is created on 2025-09-02

Relaxing Sounds
Playlist is created on 2022-07-01

Reggae Rhythms
Playlist is created on 2023-09-02

Rap Revolution
Playlist is created on 2024-08-15

Pop Perfection
Playlist is created on 2023-03-08

Party Anthems
Playlist is created on 2022-06-10

Latin Fiesta
Playlist is created on 2023-05-01

Indie Showcase
Playlist is created on 2023-01-08

Hip Hop Hype
Playlist is created on 2023-08-18

Gospel Glory
Playlist is created on 2024-09-02

Golden Classics
Playlist is created on 2022-09-02

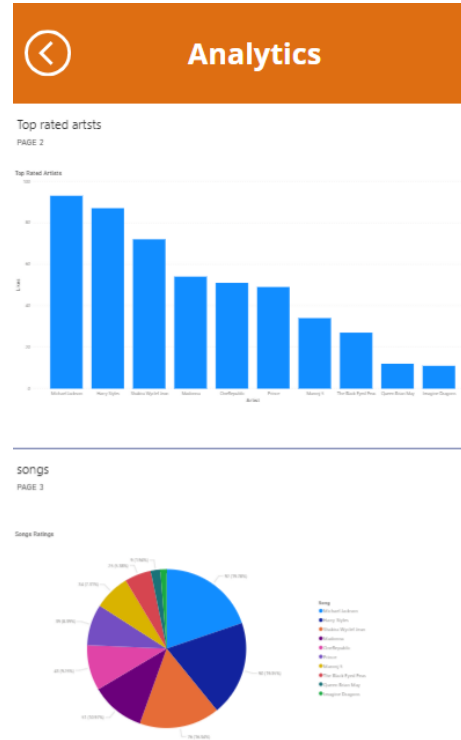
USER PROFILE

 User Profile 




Display Name


BIO

ANALYTICS



ADMIN HOME

 ADMIN  

 Search items

Baby
Artist: John Kane

All Star
Artist: Smash mash

Animal
Artist: Maronj 5

Believer
Artist: Imagine Dragons

Billie Jean
Artist: Michael Jackson

Billie Jean
Artist: Michael Jackson

Blinding Lights
Artist: The Weeknd



Bohemian Rhapsody
Artist: Queen Freddie Mercury

Boogie Wonderland
Artist: Earth Wind & Fire

Can't Stop the Feeling!
Artist: Justin Timberlake

Counting Stars
Artist: OneRepublic

ADMIN-NEW SONG


 **New Song** 

* Title of the song:


* Artist First Name:

Artist Last Name

* Duration of the song

Release Date of the song:
 

ADMIN – SONG DETAILS

 **Song Details**

Song_Title
Baby

song_Artist_firstname
John



song_Artist_lastname
Kane

Song_Duration
4.7

song_ReleaseDate
1/4/2019

Update

ADMIN – UPDATE SONG


 **Update song** 

* **Song_Title**

* **song_Artist_firstname**

song_Artist_lastname

* **Song_Duration**

song_ReleaseDate
 

SQL UP DOWN SCRIPT

```
if not exists(select * from sys.databases where name='projectnirvana')
    create database projectnirvana
GO
use projectnirvana
GO

--DOWN
--Dropping constraints
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserRoles_userrole_UserID')
    alter table UserRoles drop CONSTRAINT fk_UserRoles_userrole_UserID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserRoles_userrole_RoleID')
    alter table UserRoles drop CONSTRAINT fk_UserRoles_userrole_RoleID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_SongGenres_SongGenre_SongID')
    alter table SongGenres drop CONSTRAINT fk_SongGenres_SongGenre_SongID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_SongGenres_SongGenre_GenreID')
    alter table SongGenres drop CONSTRAINT fk_SongGenres_SongGenre_GenreID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_playlists_Playlist_CreatorID')
    alter table Playlists drop CONSTRAINT fk_playlists_Playlist_CreatorID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserInteractions_userInteraction_user')
    alter table UserInteractions drop CONSTRAINT fk_UserInteractions_userInteraction_user;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserInteractions_userInteraction_song')
    alter table UserInteractions drop CONSTRAINT fk_UserInteractions_userInteraction_song;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserInteractions_userInteraction_playlist')
    alter table UserInteractions drop CONSTRAINT fk_UserInteractions_userInteraction_playlist;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_UserProfiles_UserProfile_ID')
    alter table UserProfiles drop CONSTRAINT fk_UserProfiles_UserProfile_ID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_Analytics_Analytic_userID')
    alter table Analytics drop CONSTRAINT fk_Analytics_Analytic_userID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_Analytics_Analytic_songID')
    alter table Analytics drop CONSTRAINT fk_Analytics_Analytic_songID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_Analytics_Analytic_playlistID')
    alter table Analytics drop CONSTRAINT fk_Analytics_Analytic_playlistID;
if EXISTS(SELECT* from INFORMATION_SCHEMA. TABLE_CONSTRAINTS where
constraint_name='fk_Administrators_Administrator_ID')
    alter table Administrators drop CONSTRAINT fk_Administrators_Administrator_ID;

--dropping tables
```

```

drop table if exists Users;
drop table if exists Roles;
drop table if exists UserRoles;
drop table if exists Songs;
drop table if exists Genres;
drop table if exists SongGenres;
drop table if exists UserInteractions;
drop table if exists UserProfiles;
drop table if exists Analytics;
drop table if exists Administrators ;

--dropping stored procedure
if exists(select * from sys.objects where name = 'AddNewSong')
    drop PROCEDURE AddNewSong

--dropping views
if exists(select * from sys.objects where name='UserPlaylists')
    drop view UserPlaylists

if exists(select * from sys.objects where name='TopGenres')
    drop view TopGenres

GO

--UP METADATA
--users
CREATE TABLE Users (
    User_ID INT identity NOT NULL,
    User_FirstName NVARCHAR(255),
    User_LastName NVARCHAR(255),
    User_email NVARCHAR(255) NOT NULL,
    user_Password NVARCHAR(255) NOT NULL,
    user_RegistrationDate DATE NOT NULL,
    user_LastLoginDate DATE,
    CONSTRAINT pk_users_user_id PRIMARY KEY(User_ID),
    CONSTRAINT u_users_user_email UNIQUE (User_email),
    CONSTRAINT ck_valid_user_password CHECK (LEN(user_Password) >=4 and LEN(user_Password) <=
25)
);

--roles
CREATE TABLE Roles (
    Role_ID INT NOT NULL,
    Role_Name NVARCHAR(50) NOT NULL,
    CONSTRAINT pk_Roles_Role_ID PRIMARY KEY (Role_ID),
    CONSTRAINT u_Roles_Role_Name UNIQUE (Role_Name)
);

--UserRoles
CREATE TABLE UserRoles (
    userrole_UserID INT,

```

```

        userrole_RoleID INT,
        CONSTRAINT pk_UserRoles_userrole_UserID PRIMARY KEY (userrole_UserID, userrole_RoleID)
    );

alter table messages
    add CONSTRAINT fk_UserRoles_userrole_UserID FOREIGN key (userrole_UserID)
        REFERENCES Users (User_ID);
alter table messages
    add CONSTRAINT fk_UserRoles_userrole_RoleID FOREIGN key (userrole_RoleID)
        REFERENCES Roles (Role_ID);

--Songs
CREATE TABLE Songs (
    Song_ID INT identity NOT NULL,
    Song_Title NVARCHAR(255) NOT NULL,
    song_Artist_firstname NVARCHAR(255) NOT NULL,
    song_Artist_lastname NVARCHAR(255),
    song_ReleaseDate DATE,
    Song_Duration FLOAT NOT NULL
    CONSTRAINT pk_songs_song_id PRIMARY KEY (Song_ID)
);

--Genres
CREATE TABLE Genres (
    Genre_ID INT identity not null,
    Genre_Name NVARCHAR(255),
    CONSTRAINT pk_Genres_Genre_ID PRIMARY KEY (Genre_ID),
    CONSTRAINT u_Genres_Genre_Name UNIQUE (Genre_Name)
);

--songgenres
CREATE TABLE SongGenres (
    SongGenre_of_SongID INT not null,
    SongGenre_GenreID INT not null,
    CONSTRAINT pk_SongGenres_ID PRIMARY KEY (SongGenre_of_SongID, SongGenre_GenreID)
);

alter table SongGenres
    add CONSTRAINT fk_SongGenres_SongGenre_SongID FOREIGN key (SongGenre_of_SongID)
        REFERENCES Songs (Song_ID);
alter table SongGenres
    add CONSTRAINT fk_SongGenres_SongGenre_GenreID FOREIGN key (SongGenre_GenreID)
        REFERENCES Genres (Genre_ID);

--Playlists
CREATE TABLE Playlists (
    Playlist_ID INT identity NOT NULL,
    Playlist_Title NVARCHAR(255) NOT NULL,
    Playlist_CreatorID INT NOT NULL,
    Playlist_CreationDate DATE NOT NULL,
    Playlist_ModificationDate DATE NOT NULL,
    CONSTRAINT pk_playlists_playlist_id PRIMARY KEY (Playlist_ID)
);

```

```

alter table Playlists
    add CONSTRAINT fk_playlists_Playlist_CreatorID FOREIGN key (Playlist_CreatorID)
        REFERENCES Users (User_ID);

--userinteractions
CREATE TABLE UserInteractions (
    UserInteraction_ID INT identity NOT NULL,
    UserInteraction_Likes INT NOT NULL,
    UserInteraction_Date DATE NOT NULL,
    userInteraction_user INT,
    userInteraction_song INT,
    userinteraction_playlist INT,
    CONSTRAINT pk_userinteractions_userinteraction_id PRIMARY KEY (UserInteraction_ID)
);

alter table UserInteractions
    add CONSTRAINT fk_UserInteractions_userInteraction_user FOREIGN key (userInteraction_user)
        REFERENCES Users (User_ID);
alter table UserInteractions
    add CONSTRAINT fk_UserInteractions_userInteraction_song FOREIGN key (userInteraction_song)
        REFERENCES Songs (Song_ID);
alter table UserInteractions
    add CONSTRAINT fk_UserInteractions_userInteraction_playlist FOREIGN key
(userinteraction_playlist)
        REFERENCES Playlists (Playlist_ID);

--userprofiles
CREATE TABLE UserProfiles (
    UserProfile_ID INT NOT NULL,
    Userprofile_DisplayName NVARCHAR(255) NOT NULL,
    Userprofile_BIO NVARCHAR(MAX),
    CONSTRAINT pk_userprofiles_userprofile_id PRIMARY KEY (UserProfile_ID)
);

alter table UserProfiles
    add CONSTRAINT fk_UserProfiles_UserProfile_ID FOREIGN key (UserProfile_ID)
        REFERENCES Users (User_ID)

--Analytics
CREATE TABLE Analytics (
    Analytic_ID INT identity NOT NULL,
    Analytic_InteractionType NVARCHAR(50) NOT NULL,
    Analytic_InteractionDate DATE NOT NULL,
    Analytic_UserID INT,
    Analytic_SongID INT,
    Analytic_PlaylistID INT,
    CONSTRAINT pk_analytics_analytic_id PRIMARY KEY (Analytic_ID)
);

alter table Analytics
    add CONSTRAINT fk_Analytics_Analytic_userID FOREIGN key (Analytic_UserID)
        REFERENCES Users (User_ID);
alter table Analytics
    add CONSTRAINT fk_Analytics_Analytic_songID FOREIGN key (Analytic_SongID)

```



```

        REFERENCES Songs (Song_ID);
alter table Analytics
    add CONSTRAINT fk_Analytics_Analytic_playlistID FOREIGN key (Analytic_PlaylistID)
        REFERENCES Playlists (Playlist_ID);

--Administrators
CREATE TABLE Administrators (
    Administrator_ID INT NOT NULL,
    Administrator_Creation_date DATE,
    Administrator_roleID INT,
    Administrator_name VARCHAR(50),
    CONSTRAINT pk_Administrators_Administrator_ID PRIMARY KEY (Administrator_ID)
);

alter table Analytics
    add CONSTRAINT fk_Administrators_Administrator_ID FOREIGN key (Administrator_ID)
        REFERENCES Users (User_ID);

--VIEWS
--VIEW FOR playlist of users
CREATE VIEW UserPlaylists AS
SELECT
u.User_ID,u.User_FirstName,u.User_LastName,up.Userprofile_DisplayName,p.Playlist_ID,p.Playlist_
Title,
p.Playlist_CreationDate,p.Playlist_ModificationDate
FROM Users u
JOIN UserProfiles up ON u.User_ID = up.UserProfile_ID
LEFT JOIN Playlists p ON u.User_ID = p.Playlist_CreatorID;

--VIEW FOR TOP GENRES
CREATE VIEW TopGenres AS
SELECT
g.Genre_ID,g.Genre_Name,COUNT(sg.SongGenre_of_SongID) AS SongCount
FROM Genres g
LEFT JOIN SongGenres sg ON g.Genre_ID = sg.SongGenre_GenreID
GROUP BY g.Genre_ID, g.Genre_Name
ORDER BY SongCount DESC;

--STORED PROCEDURE TO ADD NEW SONG
CREATE PROCEDURE AddNewSong
    @Title NVARCHAR(255),
    @ArtistFirstName NVARCHAR(255),
    @ArtistLastName NVARCHAR(255),
    @ReleaseDate DATE,
    @Duration FLOAT
AS
BEGIN
    INSERT INTO Songs (Song_Title, song_Artist_firstname, song_Artist_lasttname,
song_ReleaseDate, Song_Duration)
    VALUES (@Title, @ArtistFirstName, @ArtistLastName, @ReleaseDate, @Duration);
END;

GO

```

```

--UP DATA
--USERS
INSERT INTO Users (User_FirstName, User_LastName, User_email, user_Password,
user_RegistrationDate, user_LastLoginDate)
VALUES
('John', 'Doe', 'john.doe@email.com', 'userrrr1', '2022-01-01', '2022-01-05'),
('Jane', 'Smith', 'jane.smith@email.com', 'userrrr2', '2022-02-15', '2022-02-20'),
('Robert', 'Johnson', 'robert.johnson@email.com', 'adminnnn3', '2022-03-10', '2022-03-15'),
('Alice', 'Williams', 'alice.w@email.com', 'userrrr4', '2022-04-05', '2022-04-10'),
('Charlie', 'Brown', 'charlie.b@email.com', 'userrrr5', '2022-05-20', '2022-05-25'),
('Emily', 'Davis', 'emily.d@email.com', 'userrrr6', '2022-06-12', '2022-06-18'),
('Frank', 'Johnson', 'frank.j@email.com', 'userrrr7', '2022-07-08', '2022-07-12'),
('Grace', 'Thompson', 'grace.t@email.com', 'adminnnn8', '2022-08-15', '2022-08-20'),
('Henry', 'Lee', 'henry.l@email.com', 'userrrr9', '2022-09-02', '2022-09-06'),
('Isabel', 'Clark', 'isabel.c@email.com', 'userrrr10', '2022-10-18', '2022-10-22'),
('Jack', 'Robinson', 'jack.r@email.com', 'userrrr11', '2022-11-05', '2022-11-10'),
('Kelly', 'Turner', 'kelly.t@email.com', 'userrrr12', '2022-12-12', '2022-12-18'),
('Leo', 'Evans', 'leo.e@email.com', 'adminnnn13', '2023-01-08', '2023-01-12'),
('Mia', 'Perez', 'mia.p@email.com', 'userrrr14', '2023-02-20', '2023-02-25'),
('Nathan', 'Baker', 'nathan.b@email.com', 'userrrr15', '2023-03-08', '2023-03-12'),
('Olivia', 'Fisher', 'olivia.f@email.com', 'userrrr16', '2023-04-15', '2023-04-20'),
('Peter', 'Hayes', 'peter.h@email.com', 'adminnnn17', '2023-05-01', '2023-05-05'),
('Quinn', 'Cooper', 'quinn.c@email.com', 'userrrr18', '2023-06-10', '2023-06-15'),
('Rachel', 'Lopez', 'rachel.l@email.com', 'userrrr19', '2023-07-05', '2023-07-10'),
('Samuel', 'Ward', 'samuel.w@email.com', 'userrrr20', '2023-08-18', '2023-08-22'),
('Tina', 'Long', 'tina.l@email.com', 'userrrr21', '2023-09-02', '2023-09-06'),
('Ulysses', 'Harrison', 'ulysses.h@email.com', 'adminnnn22', '2023-10-10', '2023-10-15'),
('Violet', 'Lee', 'violet.l@email.com', 'userrrr23', '2023-11-20', '2023-11-25'),
('William', 'Cruz', 'william.c@email.com', 'userrrr24', '2023-12-08', '2023-12-12'),
('Xander', 'Mitchell', 'xander.m@email.com', 'userrrr25', '2024-01-15', '2024-01-20'),
('Yasmine', 'Khan', 'yasmine.k@email.com', 'userrrr26', '2024-02-05', '2024-02-10'),
('Zane', 'Ramirez', 'zane.r@email.com', 'userrrr27', '2024-03-12', '2024-03-15'),
('Ava', 'Scott', 'ava.s@email.com', 'userrrr28', '2024-04-08', '2024-04-12'),
('Benjamin', 'Morgan', 'benjamin.m@email.com', 'adminnnn29', '2024-05-20', '2024-05-25'),
('Caroline', 'Nguyen', 'caroline.n@email.com', 'userrrr30', '2024-06-10', '2024-06-15'),
('Derek', 'Hill', 'derek.h@email.com', 'userrrr31', '2024-07-01', '2024-07-05'),
('Ella', 'Jenkins', 'ella.j@email.com', 'userrrr32', '2024-08-15', '2024-08-20'),
('Finn', 'Owens', 'finn.o@email.com', 'userrrr33', '2024-09-02', '2024-09-06'),
('Gemma', 'Cooper', 'gemma.c@email.com', 'userrrr34', '2024-10-18', '2024-10-22'),
('Hugo', 'Wright', 'hugo.w@email.com', 'adminnnn35', '2024-11-05', '2024-11-10'),
('Ivy', 'Reyes', 'ivy.r@email.com', 'userrrr36', '2024-12-12', '2024-12-18'),
('Jasper', 'Hill', 'jasper.h@email.com', 'userrrr37', '2025-01-08', '2025-01-12'),
('Kara', 'Fleming', 'kara.f@email.com', 'userrrr38', '2025-02-20', '2025-02-25'),
('Logan', 'Hunt', 'logan.h@email.com', 'userrrr39', '2025-03-08', '2025-03-12'),
('Morgan', 'Sullivan', 'morgan.s@email.com', 'userrrr40', '2025-04-15', '2025-04-20'),
('Nora', 'Kim', 'nora.k@email.com', 'adminnnn41', '2025-05-01', '2025-05-05'),
('Oscar', 'Chen', 'oscar.c@email.com', 'userrrr42', '2025-06-10', '2025-06-15'),
('Penelope', 'Wang', 'penelope.w@email.com', 'userrrr43', '2025-07-05', '2025-07-10'),
('Quincy', 'Yu', 'quincy.y@email.com', 'userrrr44', '2025-08-18', '2025-08-22'),
('Ruby', 'Li', 'ruby.l@email.com', 'userrrr45', '2025-09-02', '2025-09-06'),
('Sebastian', 'Garcia', 'sebastian.g@email.com', 'adminnnn46', '2025-10-10', '2025-10-15'),
('Tessa', 'Chang', 'tessa.c@email.com', 'userrrr47', '2025-11-20', '2025-11-25'),
('Uriah', 'Mendoza', 'uriah.m@email.com', 'userrrr48', '2025-12-08', '2025-12-12'),
('Vivian', 'Ng', 'vivian.n@email.com', 'userrrr49', '2026-01-15', '2026-01-20'),

```

```

('Wyatt', 'Nguyen', 'wyatt.n@email.com', 'userrrr50', '2026-02-05', '2026-02-10');

--ROLES
INSERT INTO Roles (Role_ID,Role_Name)
VALUES
('101','User'),
('102','Admin');

--GENRES
INSERT INTO GENRES (Genre_Name)
VALUES
('Rock'), ('Classic Rock'), ('Opera'), ('Pop'), ('Funk'),
('Folk'),('Disco'),('R&B'),('Soul'),('New Wave'),('Dance'),('Latin'),('Hip Hop'),('Country'),
('Alternative'),('Grunge');

--SONGS
INSERT INTO Songs (Song_Title, song_Artist_firstname, song_Artist_lastname, song_ReleaseDate,
Song_Duration)
VALUES
('Imagine', 'John', 'Lennon', '1971-09-09', 3.03),
('Bohemian Rhapsody', 'Queen', 'Freddie Mercury', '1975-10-31', 5.55),
('Billie Jean', 'Michael', 'Jackson', '1983-01-02', 4.54),
('Hotel California', 'Eagles', NULL, '1977-12-08', 6.30),
('Like a Rolling Stone', 'Bob', 'Dylan', '1965-07-20', 6.13),
('Stairway to Heaven', 'Led', 'Zeppelin', '1971-11-08', 8.02),
('Thriller', 'Michael', 'Jackson', '1982-11-30', 5.57),
('Boogie Wonderland', 'Earth', 'Wind & Fire', '1979-03-20', 4.50),
('Sweet Child o Mine', 'Guns', 'Roses', '1987-07-21', 5.55),
('Dancing Queen', 'ABBA', NULL, '1976-08-15', 3.51),
('Paint It Black', 'Rolling', 'Dice', '1966-05-07', 3.45),
('I Will Always Love You', 'Whitney', 'Houston', '1992-11-03', 4.32),
('Let It Be', 'Mark', 'Beatles', '1970-03-06', 3.50),
('Wannabe', 'Spice', 'Girls', '1996-07-08', 2.52),
('Billie Jean', 'Michael', 'Jackson', '1983-01-02', 4.54),
('Blinding Lights', 'The', 'Weeknd', '2019-11-29', 3.22),
('Bohemian Rhapsody', 'Queen', 'Freddie Mercury', '1975-10-31', 5.55),
('Despacito', 'Luis', 'Fonsi', '2017-01-12', 3.48),
('Dont Stop Believing', 'Journey', NULL, '1981-06-03', 4.10),
('Every Breath You Take', 'The', 'Police', '1983-05-20', 4.13),
('Havana', 'Camila', 'Cabello', '2017-08-03', 3.37),
('I Gotta Feeling', 'The', 'Black Eyed Peas', '2009-06-16', 4.49),
('Like a Prayer', 'Madonna', NULL, '1989-03-03', 5.41),
('Old Town Road', 'Lil', 'Nas X', '2018-12-03', 1.53),
('Rolling in the Deep', 'Adele', NULL, '2010-11-29', 3.48),
('Shape of You', 'Ed', 'Sheeran', '2017-01-06', 3.53),
('Sicko Mode', 'Travis', 'Scott', '2018-08-21', 5.12),
('Someone You Loved', 'Lewis', 'Capaldi', '2018-11-08', 3.02),
('Sweet Caroline', 'Neil', 'Diamond', '1969-05-28', 3.23),
('Uptown Funk', 'Mark', 'Ronson', '2014-11-10', 4.30),
('WAP', 'Cardi', 'B', '2020-08-07', 3.07),
('Watermelon Sugar', 'Harry', 'Styles', '2019-11-16', 2.54),
('We Will Rock You', 'Queen', 'Brian May', '1977-10-07', 2.02),
('What's Going On', 'Marvin', 'Gaye', '1971-05-21', 3.53),
('When Doves Cry', 'Prince', NULL, '1984-05-16', 5.54),

```

```

('Yellow Submarine', 'The', 'Beatles', '1966-08-05', 2.38),
('You Give Love a Bad Name', 'Bon', 'Jovi', '1986-07-01', 3.42),
('Zombie', 'The', 'Cranberries', '1994-09-19', 5.06),
('Africa', 'Toto', NULL, '1982-04-30', 4.58),
('Dance Monkey', 'Tones', 'and I', '2019-05-10', 3.29),
('Hips Don''t Lie', 'Shakira', 'Wyclef Jean', '2006-02-28', 3.39),
('Mr. Brightside', 'The', 'Killers', '2003-09-29', 3.42),
('Smells Like Teen Spirit', 'Nirvana', NULL, '1991-09-10', 5.01),
('Shape of You', 'Ed', 'Sheeran', '2017-01-06', 3.53),
('Believer', 'Imagine', 'Dragons', '2017-02-01', 3.23),
('Counting Stars', 'OneRepublic', NULL, '2013-06-14', 4.17),
('Dancing in the Dark', 'Bruce', 'Springsteen', '1984-05-03', 4.05),
('All Star', 'Smash', 'Mouth', '1999-05-04', 3.21),
('Can''t Stop the Feeling!', 'Justin', 'Timberlake', '2016-05-06', 3.56),
('Hey Jude', 'The', 'Beatles', '1968-08-26', 7.11);

--PLAYLISTS
INSERT INTO Playlists (Playlist_Title, Playlist_CreatorID, Playlist_CreationDate,
Playlist_ModificationDate)
VALUES
('Party Anthems', 6, '2022-06-10', '2022-06-15'),
('Relaxing Sounds', 7, '2022-07-01', '2022-07-05'),
('90s Throwback', 8, '2022-08-15', '2022-08-20'),
('Golden Classics', 9, '2022-09-02', '2022-09-06'),
('Indie Showcase', 13, '2023-01-08', '2023-01-12'),
('Electronic Beats', 14, '2023-02-20', '2023-02-25'),
('Pop Perfection', 15, '2023-03-08', '2023-03-12'),
('Country Hits', 16, '2023-04-15', '2023-04-20'),
('Latin Fiesta', 17, '2023-05-01', '2023-05-05'),
('Hip Hop Hype', 20, '2023-08-18', '2023-08-22'),
('Reggae Rhythms', 21, '2023-09-02', '2023-09-06'),
('Classical Serenity', 22, '2023-10-10', '2023-10-15'),
('Arabic Nights', 30, '2024-06-10', '2024-06-15'),
('Folk Fables', 31, '2024-07-01', '2024-07-05'),
('Rap Revolution', 32, '2024-08-15', '2024-08-20'),
('Gospel Glory', 33, '2024-09-02', '2024-09-06'),
('Experimental Beats', 39, '2025-03-08', '2025-03-12'),
('Dubstep Delights', 40, '2025-04-15', '2025-04-20'),
('Salsa Sensation', 45, '2025-09-02', '2025-09-06'),
('Bluegrass Bliss', 46, '2025-10-10', '2025-10-15'),
('Fusion Frenzy', 47, '2025-11-20', '2025-11-25');

--USERINTERACTIONS
INSERT INTO UserInteractions (UserInteraction_Type, userInteraction_user, userInteraction_song,
userinteraction_playlist, UserInteraction_Date)
VALUES
('Like', 8, 2, 3, '2022-01-20'),
('Like', 20, 7, 10, '2022-02-10'),
('Like', 47, 15, 21, '2022-03-18'),
('Like', 4, 23, NULL, '2022-04-05'),
('Like', 5, 32, NULL, '2022-05-22'),
('Like', 7, 35, 2, '2023-05-01'),
('Like', 39, 41, 17, '2023-06-10'),
('Like', 15, 12, 7, '2023-07-05'),

```

```

('Like', 10, 27, NULL, '2023-08-18'),
('Like', 6, 22, 1, '2024-01-15'),
('Like', 30, 33, 13, '2024-02-05'),
('Like', 45, 46, 19, '2024-03-12'),
('Like', 17, 1, 9, '2024-04-08'),
('Like', 34, 18, NULL, '2024-10-18'),
('Like', 33, 25, 16, '2024-11-05'),
('Like', 40, 34, 18, '2024-12-12'),
('Like', 9, 45, 4, '2025-01-08'),
('Like', 38, 3, NULL, '2025-02-20'),
('Like', 11, 23, NULL, '2025-10-10'),
('Like', 27, 32, NULL, '2025-11-20'),
('Like', 29, 41, NULL, '2025-12-08');

--USERPROFILES
INSERT INTO UserProfiles (UserProfile_ID, Userprofile_DisplayName, Userprofile_BIO)
VALUES
(1, 'John Doe', 'Music lover and tech enthusiast.'),
(2, 'Jane Smith', 'Passionate about discovering new tunes.'),
(3, 'Robert Johnson', 'Student who enjoys studying with music.'),
(4, 'Alice Williams', 'Aspiring artist and songwriter.'),
(5, 'Charlie Brown', 'Fitness freak who loves workout playlists.'),
(6, 'Emily Davis', 'Party planner and music enthusiast.'),
(7, 'Frank Jhonson', 'Relaxing sounds for a stressful day.'),
(8, 'Grace Thompson', '90s music nostalgia.'),
(9, 'Henry Lee', 'Classic rock is my go-to genre.'),
(10, 'Isabel Clark', 'Energetic hits for a productive day.'),
(11, 'Jack Robinson', 'Acoustic vibes and coffee lover.'),
(12, 'Kelly Turner', 'Soulful melodies soothe the soul.'),
(13, 'Leo Evans', 'Indie music explorer.'),
(14, 'Mia Perez', 'Electronic beats for the win.'),
(15, 'Nathan Baker', 'Pop perfection all day, every day.'),
(16, 'Olivia Fisher', 'Country hits and cowboy dreams.'),
(17, 'Peter Hayes', 'Latin rhythms for dance enthusiasts.'),
(18, 'Quinn Cooper', 'Jazz lover and nighttime thinker.'),
(19, 'Rachel Lopez', 'Rock classics never get old.'),
(20, 'Samuel Ward', 'Hip hop hype to keep me going.'),
(21, 'Tina Long', 'Reggae rhythms and good vibes.'),
(22, 'Ulysses Harrison', 'Classical serenity for a peaceful mind.'),
(23, 'Violet Lee', 'Indie showcase for the free spirit.'),
(24, 'William Cruz', 'Metalhead for life.'),
(25, 'Xander Mitchell', 'K-Pop sensations keep me dancing.'),
(26, 'Yasmin Khan', 'Funky grooves and positive vibes.'),
(27, 'Zane Ramirez', 'African beats for cultural connections.'),
(28, 'Ava Scott', 'Caribbean carnival for tropical vibes.'),
(29, 'Benjamin Morgan', 'Bollywood beats and Indian tunes.'),
(30, 'Caroline Nguyen', 'Arabic nights for a cultural journey.'),
(31, 'Derek Hill', 'Folk fables and storytelling melodies.'),
(32, 'Ella Jenkins', 'Rap revolution and urban beats.'),
(33, 'Finn Owens', 'Gospel glory and spiritual upliftment.'),
(34, 'Gemma Cooper', 'Choral harmony for a soothing experience.'),
(35, 'Hugo Wright', 'Piano passion and classical melodies.'),
(36, 'Ivy Reyes', 'Vocal virtuosos and powerful voices.'),
(37, 'Jasper Hill', 'Tropical tunes for a beach escape.'),

```

```

(38, 'Kara Fleming', 'Ambient dreams and calming sounds.'),
(39, 'Logan Hunt', 'Experimental beats and avant-garde music.'),
(40, 'Morgan Sullivan', 'Dubstep delights and bass drops.'),
(41, 'Nora Kim', 'Techno triumphs for the techno heads.'),
(42, 'Oscar Chen', 'House hits and electronic dance music.'),
(43, 'Penelope Wang', 'Trance treasures and euphoric beats.'),
(44, 'Quincy Yu', 'R&B revelations and soulful sounds.'),
(45, 'Ruby Li', 'Salsa sensation and Latin rhythms.'),
(46, 'Sebastian Garcia', 'Bluegrass bliss and country vibes.'),
(47, 'Tessa Chang', 'Fusion frenzy for a genre-bending experience.'),
(48, 'Uriah Mendoza', 'Swing symphony and jazz classics.'),
(49, 'Vivian Ng', 'Opera opulence and classical excellence.'),
(50, 'Wyatt Nguyen', 'Nature soundscape for relaxation and meditation.');
```

```

--ANALYTICS
INSERT INTO Analytics (Analytic_InteractionType, Analytic_UserID, Analytic_SongID,
Analytic_PlaylistID, Analytic_InteractionDate)
VALUES
('Like', 8, 2, 3, '2022-01-20'),
('Like', 20, 7, 10, '2022-02-10'),
('Like', 47, 15, 21, '2022-03-18'),
('Like', 4, 23, NULL, '2022-04-05'),
('Like', 5, 32, NULL, '2022-05-22'),
('Like', 7, 35, 2, '2023-05-01'),
('Like', 39, 41, 17, '2023-06-10'),
('Like', 15, 12, 7, '2023-07-05'),
('Like', 10, 27, NULL, '2023-08-18'),
('Like', 6, 22, 1, '2024-01-15'),
('Like', 30, 33, 13, '2024-02-05'),
('Like', 45, 46, 19, '2024-03-12'),
('Like', 17, 1, 9, '2024-04-08'),
('Like', 34, 18, NULL, '2024-10-18'),
('Like', 33, 25, 16, '2024-11-05'),
('Like', 40, 34, 18, '2024-12-12'),
('Like', 9, 45, 4, '2025-01-08'),
('Like', 38, 3, NULL, '2025-02-20'),
('Like', 11, 23, NULL, '2025-10-10'),
('Like', 27, 32, NULL, '2025-11-20'),
('Like', 29, 41, NULL, '2025-12-08');
```

```

--ADMINISTRATORS
INSERT INTO Administrators
(Administrator_Id, Administrator_Creation_date, Administrator_roleID, Administrator_name)
VALUES
(3, '2022-03-10', 102, 'Robert Johnson'),
(8, '2022-08-15', 102, 'Grace Thompson'),
(13, '2023-01-08', 102, 'Leo Evans'),
(17, '2023-05-01', 102, 'Peter Hayes'),
(22, '2023-10-10', 102, 'Ulysses Harrison');
```

```

--USERROLES
INSERT INTO UserRoles (userrole_UserID, userrole_RoleID)
Values
```

```
(1,101),  
(2,101),  
(3,102),  
(4,101),  
(5,101),  
(6,101),  
(7,101),  
(8,102),  
(9,101),  
(10,101),  
(11,101),  
(12,101),  
(13,102),  
(14,101),  
(15,101),  
(16,101),  
(17,102),  
(18,101),  
(19,101),  
(20,101),  
(21,101),  
(22,102),  
(23,101),  
(24,101),  
(25,101),  
(26,101),  
(27,101),  
(28,101),  
(29,101),  
(30,101),  
(31,101),  
(32,101),  
(33,101),  
(34,101),  
(35,101),  
(36,101),  
(37,101),  
(38,101),  
(39,101),  
(40,101),  
(41,101),  
(42,101),  
(43,101),  
(44,101),  
(45,101),  
(46,101),  
(47,101),  
(48,101),  
(49,101),  
(50,101);
```

```
--SONGENRES
```

```
INSERT INTO SongGenres (SongGenre_of_SongID,SongGenre_GenreID)  
VALUES
```

(1,16),
(1,17),
(2,16),
(2,18),
(3,19),
(3,20),
(4,16),
(5,16),
(5,21),
(6,16),
(7,19),
(7,20),
(8,21),
(8,22),
(9,16),
(10,19),
(10,22),
(11,16),
(12,23),
(12,24),
(13,16),
(14,19),
(15,19),
(15,20),
(16,19),
(16,23),
(17,18),
(17,16),
(18,19),
(19,16),
(20,25),
(20,16),
(21,19),
(21,27),
(22,19),
(22,26),
(23,19),
(24,28),
(24,29),
(25,19),
(25,24),
(26,19),
(27,28),
(28,19),
(29,16),
(29,19),
(30,19),
(30,20),
(31,28),
(32,19),
(33,16),
(34,23),
(34,24),
(35,19),


```

(35,20),
(36,16),
(37,16),
(38,30),
(39,16),
(40,19),
(41,19),
(41,27),
(42,16),
(43,31),
(44,19),
(45,16),
(45,19),
(46,19),
(46,16),
(47,16),
(48,19),
(48,16),
(49,19),
(50,16);

GO

--VERIFY
SELECT * from Users;
SELECT * from Roles;
SELECT * from UserRoles;
SELECT * from Songs;
SELECT * from Genres;
SELECT * from SongGenres;
SELECT * from Playlists;
SELECT * from UserInteractions;
SELECT * from UserProfiles;
SELECT * from Analytics;
SELECT * from Administrators;
GO

--TO DISPLAY TOP RATED ARTISTS
WITH ArtistRatings AS (
SELECT
sa.song_Artist_firstname + ' ' + ISNULL(sa.song_Artist_lastname, '') AS ArtistFullName,
DENSE_RANK() OVER (ORDER BY COUNT(ui.UserInteraction_ID) DESC) AS ArtistRank
FROM Songs sa
JOIN UserInteractions ui ON sa.Song_ID = ui.userInteraction_song
GROUP BY sa.song_Artist_firstname, sa.song_Artist_lastname
)
SELECT ArtistFullName,ArtistRank
FROM ArtistRatings
WHERE ArtistRank <= 3;

--TO DISPLAY TOP RATED SONGS
WITH SongRatings AS (
SELECT
sa.Song_Title,
sa.song_Artist_firstname + ' ' + ISNULL(sa.song_Artist_lastname, '') AS ArtistFullName,

```

```

DENSE_RANK() OVER (ORDER BY COUNT(ui.UserInteraction_ID) DESC) AS SongRank
FROM Songs sa
JOIN UserInteractions ui ON sa.Song_ID = ui.userInteraction_song
GROUP BY sa.Song_ID, sa.Song_Title, sa.song_Artist_firstname, sa.song_Artist_lastname
)
SELECT Song_Title, ArtistFullName, SongRank
FROM SongRatings
WHERE SongRank <= 3;

--STORED PROCEDURE TO ADD NEW SONG
CREATE PROCEDURE AddNewSong
    @Title NVARCHAR(255),
    @ArtistFirstName NVARCHAR(255),
    @ArtistLastName NVARCHAR(255),
    @ReleaseDate DATE,
    @Duration FLOAT
AS
BEGIN
    INSERT INTO Songs (Song_Title, song_Artist_firstname, song_Artist_lastname,
song_ReleaseDate, Song_Duration)
    VALUES (@Title, @ArtistFirstName, @ArtistLastName, @ReleaseDate, @Duration);
END;

EXEC AddNewSong
    @Title = ' Baby',
    @ArtistFirstName = 'John',
    @ArtistLastName = 'Kane',
    @ReleaseDate = '2019-01-04',
    @Duration = '4.7';

```

TIME LOG OF TEAMMATES

Leena Balagalu

18/11/23 – Both of us has analyzed metadata, data, entities, attributes, relationships and cardinality and divided 3 tables each for creating spreadsheet.

19/11/23 – Both of us have created excel with allocated tables.

20/11/23 – Reviewed our excel sheets which were individually created.

24/11/23 – Discussed about conceptual and logical models.

25/11/23 – Created conceptual model for the project.

26/11/23 – discussed on creating tables and individual data sets for insertion.

27/11/23 – Modified the logical model created by Sindhuri and created all the tables using azure data studio.

28/11/23 – Created datasets manually and using mockaroo for 5 Tables.

29/11/23 – Created datasets manually and using mockaroo for remaining 6 Tables.

30/11/23 – Inserted data in all of the tables.

2/12/23 – Started PowerApps and completed designing 4 screens.

3/12/23 – Completed designing all of the remaining screens in PowerApps.

4/12/23 – Helped in creating presentation.

Demonstrated final team presentation.

9/12/23 – Modified the PowerApps with hashing of password and integrating powerbi with power bi tile.

10/12/23 – Project Report.

Sai Sinduri Vangala

18/11/23 – Both of us has analyzed metadata, data, entities, attributes , relationships and cardinality and divided 3 tables each for creating spreadsheet.

19/11/23 – Both of us have created excel with allocated tables.

20/11/23 – Reviewed our excel sheets which were individually created.

24/11/23 – Discussed regarding conceptual and logical models.

25/11/23 – created logical model.

26/11/23 – discussed on creating tables and individual data sets for insertion.

3/11/23 – Created use cases for the project.

4/11/23 – created the final presentation.

Demonstrated final team presentation.

PRESENTATION SLIDES WITH DEMO

[nirvanafinal.pptx](#)

-