Seed Data (Group 05)

**Link :** [**https://github.com/ShivaniPandula/MusicMix\_GDP-05/wiki/Seed-Data**](https://github.com/ShivaniPandula/MusicMix_GDP-05/wiki/Seed-Data)

-- Create the database (assuming it's not already created)

CREATE DATABASE music\_mix;

-- Use the created database

USE music\_mix;

-- Create User table

CREATE TABLE User (

UserId INT PRIMARY KEY,

SongId INT,

UserName VARCHAR(255),

EmailID VARCHAR(255),

Password VARCHAR(255),

Gender VARCHAR(10),

DOB DATE,

PhoneNumber VARCHAR(20)

);

-- Insert data into User table

INSERT INTO User (UserId, SongId, UserName, EmailID, Password, Gender, DOB, PhoneNumber) VALUES

(1001, 2001, 'shivani', 'shivani@example.com', 'hashedpassword1', 'Female', TO\_DATE('1990-01-01', 'YYYY-MM-DD'), '1234567890'),

(1002, 2002, 'shivaram', 'shivaram@example.com', 'hashedpassword2', 'Male', TO\_DATE('1992-02-02', 'YYYY-MM-DD'), '0987654321'),

(1003, 2003, 'srinadh', 'srinadh@example.com', 'hashedpassword3', 'Male', TO\_DATE('1988-03-03', 'YYYY-MM-DD'), '1112223333'),

(1004, 2004, 'sayendra', 'sayendra@example.com', 'hashedpassword4', 'Male', TO\_DATE('1995-04-04', 'YYYY-MM-DD'), '4445556666'),

(1005, 2005, 'rajkumar', 'rajkumar@example.com', 'hashedpassword5', 'Male', TO\_DATE('1987-05-05', 'YYYY-MM-DD'), '7778889999'),

(1006, 2006, 'satwik', 'satwik@example.com', 'hashedpassword6', 'Male', TO\_DATE('1993-06-06', 'YYYY-MM-DD'), '1010101010');

-- Create Playlist table

CREATE TABLE Playlist (

PlaylistId INT PRIMARY KEY,

UserId INT,

Details VARCHAR(255),

FOREIGN KEY (UserId) REFERENCES User(UserId)

);

-- Insert data into Playlist table

INSERT INTO Playlist (PlaylistId, UserId, Details) VALUES

(3001, 1001, 'Shivani’s Rock Playlist'),

(3002, 1002, 'Shivaram’s Pop Playlist'),

(3003, 1003, 'Srinadh’s Playlist'),

(3004, 1004, 'Sayendra’s Playlist'),

(3005, 1005, 'Rajkumar’s Playlist'),

(3006, 1006, 'Satwik’s Playlist');

-- Create Song table

CREATE TABLE Song (

SongId INT PRIMARY KEY,

AlbumId INT,

ArtistId INT,

SongName VARCHAR(255),

Genre VARCHAR(50),

Lyrics VARCHAR(1000),

Duration INT,

FOREIGN KEY (AlbumId) REFERENCES Album(AlbumId),

FOREIGN KEY (ArtistId) REFERENCES Artist(ArtistId)

);

-- Insert data into Song table

INSERT INTO Song (SongId, AlbumId, ArtistId, SongName, Genre, Lyrics, Duration) VALUES

(2001, 4001, 5001, 'cheri cheri', 'Rock', 'Lyrics of cheri cheri', 180),

(2002, 4002, 5002, 'darshana', 'Pop', 'Lyrics of darshana', 200);

-- Create Album table

CREATE TABLE Album (

AlbumId INT PRIMARY KEY,

AlbumName VARCHAR(255),

ReleaseDate DATE

);

-- Insert data into Album table

INSERT INTO Album (AlbumId, AlbumName, ReleaseDate) VALUES

(4001, 'Rock Album', TO\_DATE('2020-01-01', 'YYYY-MM-DD')),

(4002, 'Pop Album', TO\_DATE('2021-01-01', 'YYYY-MM-DD'));

-- Create Artist table

CREATE TABLE Artist (

ArtistId INT PRIMARY KEY,

ArtistName VARCHAR(255),

Bio VARCHAR(1000)

);

-- Insert data into Artist table

INSERT INTO Artist (ArtistId, ArtistName, Bio) VALUES

(5001, 'pavan', 'Bio of pavan'),

(5002, 'hari', 'Bio of hari');

-- Create Admin table

CREATE TABLE Admin (

AdminId INT PRIMARY KEY,

UserName VARCHAR(255),

Password VARCHAR(255)

);

-- Insert data into Admin table

INSERT INTO Admin (AdminId, UserName, Password) VALUES

(6001, 'admin', 'hashedadminpassword');

-- Create SongArtist table

CREATE TABLE SongArtist (

SongArtistId INT PRIMARY KEY,

SongId INT,

ArtistId INT,

FOREIGN KEY (SongId) REFERENCES Song(SongId),

FOREIGN KEY (ArtistId) REFERENCES Artist(ArtistId)

);

-- Insert data into SongArtist table

INSERT INTO SongArtist (SongArtistId, SongId, ArtistId) VALUES

(7001, 2001, 5001),

(7002, 2002, 5002);

-- Create AlbumArtist table

CREATE TABLE AlbumArtist (

AlbumArtistId INT PRIMARY KEY,

AlbumId INT,

ArtistId INT,

FOREIGN KEY (AlbumId) REFERENCES Album(AlbumId),

FOREIGN KEY (ArtistId) REFERENCES Artist(ArtistId)

);

-- Insert data into AlbumArtist table

INSERT INTO AlbumArtist (AlbumArtistId, AlbumId, ArtistId) VALUES

(8001, 4001, 5001),

(8002, 4002, 5002);

-- Create SongPlaylist table

CREATE TABLE SongPlaylist (

SongPlaylistId INT PRIMARY KEY,

SongId INT,

PlaylistId INT,

FOREIGN KEY (SongId) REFERENCES Song(SongId),

FOREIGN KEY (PlaylistId) REFERENCES Playlist(PlaylistId)

);

-- Insert data into SongPlaylist table

INSERT INTO SongPlaylist (SongPlaylistId, SongId, PlaylistId) VALUES

(9001, 2001, 3001),

(9002, 2002, 3002);

-- Create Feedback table

CREATE TABLE Feedback (

FeedbackID INT PRIMARY KEY,

FeedbackText VARCHAR(1000),

UserId INT,

FOREIGN KEY (UserId) REFERENCES User(UserId)

);

-- Insert data into Feedback table

INSERT INTO Feedback (FeedbackID, FeedbackText, UserId) VALUES

(10001, 'Great app!', 1001),

(10002, 'Needs improvement.', 1002),

(10003, 'Nice features!', 1003),

(10004, 'Good work.', 1004),

(10005, 'Amazing app!', 1005),

(10006, 'User-friendly interface.', 1006);

## **Instructions for loading the test data into the database:**

### Clone the Repository

Ensure you have the latest version of the repository cloned on your local machine:

git clone <https://github.com/ShivaniPandula/MusicMix_GDP-05.git>

cd MusicMix\_GDP-05

### Database Setup

Ensure your database is set up and running. For this example, we'll assume you are using MySQL.

Create a new database for testing:

CREATE DATABASE musicmix\_test;

### Import the Schema: Import the SQL schema file to set up the necessary tables:

mysql -u your\_username -p musicmix\_test < music\_mix.sql

Replace your\_username with your MySQL username. You will be prompted to enter your MySQL password after running this command.

### Load Seed Data: Load the seed data into the database:

mysql -u your\_username -p musicmix\_test < music\_mix.sql