NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL

Academic year-2025-2026

Prepared By : Ashwattha Bandhaokar
(23CEB0B08)

Subject : Music Streaming App Database

SYNOPSIS

Problem statement :

Developing a robust and efficient database management system (DBMS) for a **music streaming platform**, aiming to :

1. Data-storage optimization
2. enhance user experience
3. provide valuable insights to Artists
4. Help to improve recommendation algorithms
5. Enhancing overall user experience

The system will encompass various entity sets, including songs, artists,users, playlists, genres, and user interactions.
This Database also can be used for optimizing performance and security.
By analyzing user interactions and preferences, the DBMS can provide valuable insights to artists, allowing them to understand which types of songs are gaining popularity and where their music is going viral. This information empowers artists to produce better-quality songs and tailor their content to suit audience tastes, ultimately enhancing their visibility and success in the industry.

Moreover, the database can facilitate targeted advertising based on user preferences, enabling advertisers to reach their desired audience segments effectively.

The goal is to create a **scalable and reliable database system** that supports the platform's growth and delivers a seamless and enjoyable music streaming experience to users worldwide.

Entity involved in ER Model :

1. User : It stores user information.It is a strong entity set having user_id,User_name,User_age,mob_no,state,Nation.

2. Song : it stores the data of a particular song. It is a strong entity set.It contains Song_id, song_name, MP3_song, Lyrics and date_of_upload.

3. Artist : It stores the data of artists. It is a strong entity set.It contains Artist_id,Artist_name, State, Nationality, Gender, age. It also has 3 IS A relationship with Lyricist, Singer, Music_composer.

1. Lyricist : It contains the writing style of a Lyricist.
2. Music Composer : contains music_style and instrumentation.
3. Singer : contains Vocal_type and singing_style.

4. Artist_type : It contains the artist type key and Its type.It is a strong entity set.

5. Language : It stores the language details.It is a strong entity set.It contains Language_id,Language_name.

6. Genre : It stores information about the genre of music. It is a strong entity set.It includes genre_id,genre_name.

7. Playlist : A table that contains information about playlists in the system made by the user. It is a strong entity set as a playlist_id is the primary key in the entity set and does not depend on any other set. It includes playlist_id, Playlist_name. It has relationships with User who made it and also with the songs in it.

8. Subscription : It contains information about subscription plans available in our app. This is a strong entity set. It contains subscription_plan_id, plan_duration, plan_type, plan_amount.

9. Payment_history : It contains the information about payment done through the app to buy a subscription.It is a Weak entity set as depends on User_id for key. It includes payment_id, paid_amount, mode of payment and payment_date.

10. Customer_service : It contains the information about the customer service transactions. It is a weak entity set as it requires user_id in its key which is a foreign key. It contains service_id,user_id, date_of_service and problem.

Assumption in ER Model :

1. Some obvious assumptions are 1 user can play and like multiple songs and playlists.1 playlist can have multiple songs.one user can do multiple payments.
2. A song has only one language (Major Language is considered).
3. 1 song can have multiple artists each of lyricist, music_composer and singer. Also a song can have only 1 genre (such as rock,jazz,classical,love,etc.) but a genre can have multiple songs in it.
4. A playlist can have only One creator.
5. A user can purchase multiple subscriptions.
6. A user can select only 1 preferred Language.
7. Subscription plan duration is considered in months (1 month / 2 month / 3 month …)

About Relationships :

User →[played]→ Song :
one user can listen to multiple songs and one song can be listened to by many users.
➔ Cardinality : Many to Many

User →[Liked]→ Song :
one user can like multiple songs and one song can be liked by many users
➔ Cardinality : Many to Many

User →[serviced_by]→ Customer_service :
one user can have problems multiple times and can be serviced multiple times.
➔ Cardinality : One to Many

User →[created]→ Playlist :
One user can create multiple playlists but one playlist can be created by a single user.
➔ Cardinality : One to Many

User →[Liked]→ Playlist :
One user can like multiple playlists and one playlist can be liked by multiple users.
➔ Cardinality : Many to Many

User →[Purchase]→ Subscription_plan :
One user can get multiple plans and multiple users can get the same plan.
➔ Cardinality : Many to Many

User →[Prefer]→ Language :
One user can select only one language but a language can be selected by multiple users.
➔ Cardinality : Many to One

Payment_history →[Done By] → User :
One user can make multiple payments.
➔ Cardinality : Many to One

Playlist →[Has]→ Song :
One Playlist can have multiple songs and One song Can be in multiple playlists.
➔ Cardinality : Many to Many

Artist →[Has]→ Artist_type :
One artist is of a type but a type can have multiple artists (Multiple artists of the same type as singer/lyricist/…).
➔ Cardinality : Many to One

Song →[Written By]→ Lyricist :
One Song can be written by 1 Lyricist (assumption) but 1 Lyricist can Write multiple songs.
➔ Cardinality : Many to One

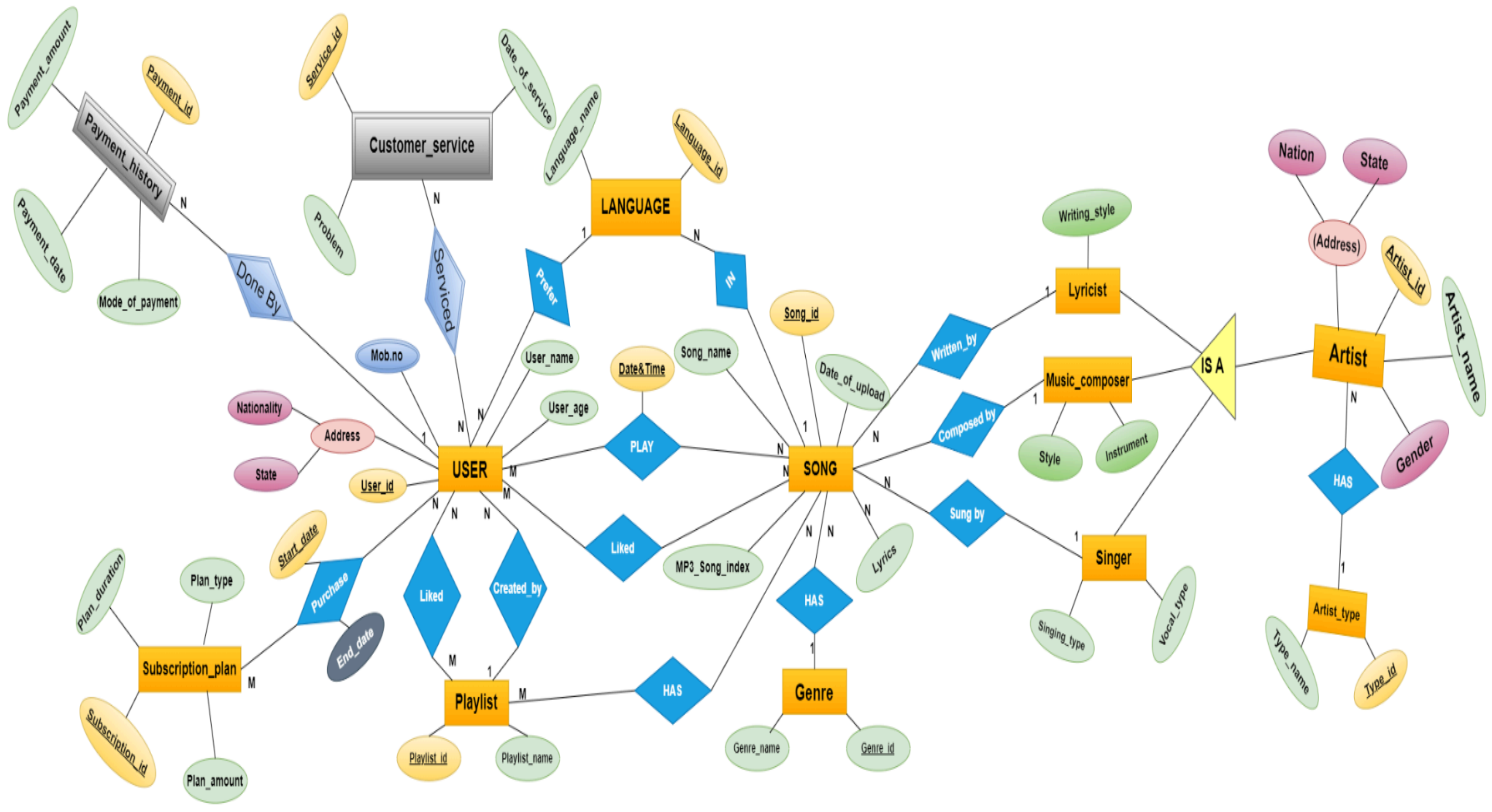
Song →[Sung By]→ Singer :
One Song can Sung by One Singer (assumption) but One Singer can Sing multiple songs.
➔ Cardinality : Many to One

Song →[Composed By]→ Composer :
One Song is Composed by One Composer (assumption) but One Composer can Compose multiple songs.
➔ Cardinality : Many to One

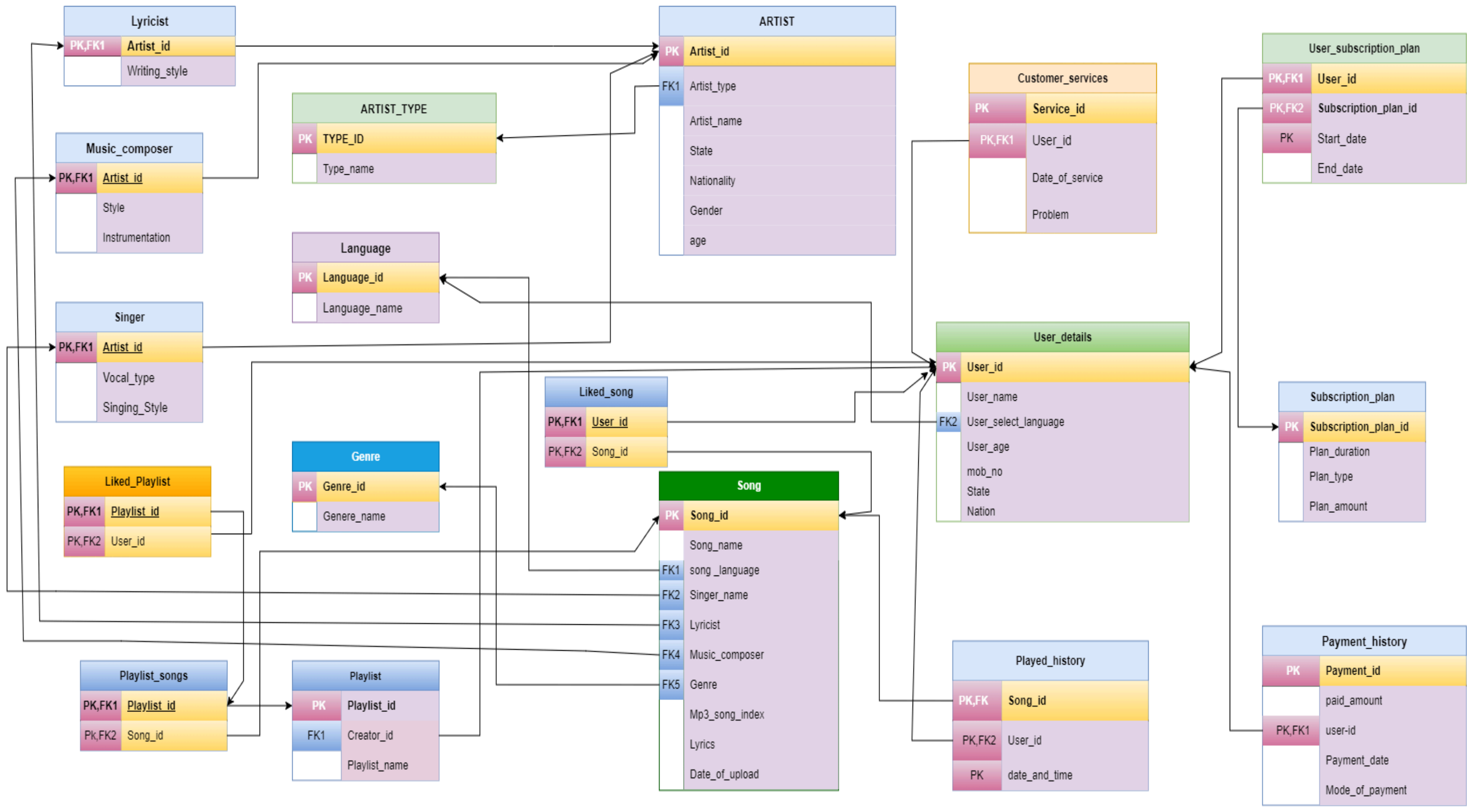
Song →[Has]→ Genre :
One song has one genre (rough assumption) but one genre can have multiple songs in it.
➔ Cardinality : Many to One

Language →[IN]→ Song :
There can be many songs in a language but a song can have 1 major language(assumption).
➔ Cardinality : Many to One

ER Diagram :


Payment_amount
Payment_id
Payment_history
Payment_date
Mode_of_payment
Done By
Service_id
Customer_service
Date_of_service
Problem
Serviced
Language_name
Language_id
LANGUAGE
Prefer
IN
Song_id
Mob.no
User_name
User_age
Nationality
Address
State
User_id
USER
Date&Time
PLAY
Song_name
Date_of_upload
SONG
MP3_Song_index
Lyrics
Liked
Written_by
Lyricist
Writing_style
Composed by
Music_composer
Style
Instrument
Sung by
Singer
Singing_type
Vocal_type
IS A
Nation
State
(Address)
Artist_id
Artist
Artist_name
Gender
HAS
Artist_type
Type_name
Type_id
Plan_duration
Plan_type
Subscription_plan
Subscription_id
Plan_amount
Start_date
Purchase
End_date
Liked
Created_by
Playlist
Playlist_id
Playlist_name
HAS
HAS
Genre
Genre_name
Genre_id

ER schema


Lyricist
PK,FK1
Artist_id
Writing_style
ARTIST
PK
Artist_id
FK1
Artist_type
Artist_name
State
Nationality
Gender
age
ARTIST_TYPE
PK
TYPE_ID
Type_name
Music_composer
PK,FK1
Artist_id
Style
Instrumentation
Language
PK
Language_id
Language_name
Singer
PK,FK1
Artist_id
Vocal_type
Singing_Style
Customer_services
PK
Service_id
PK,FK1
User_id
Date_of_service
Problem
User_subscription_plan
PK,FK1
User_id
PK,FK2
Subscription_plan_id
PK
Start_date
End_date
User_details
PK
User_id
User_name
FK2
User_select_language
User_age
mob_no
State
Nation
Subscription_plan
PK
Subscription_plan_id
Plan_duration
Plan_type
Plan_amount
Liked_song
PK,FK1
User_id
PK,FK2
Song_id
Genre
PK
Genre_id
Genere_name
Liked_Playlist
PK,FK1
Playlist_id
PK,FK2
User_id
Song
PK
Song_id
Song_name
FK1
song _language
FK2
Singer_name
FK3
Lyricist
FK4
Music_composer
FK5
Genre
Mp3_song_index
Lyrics
Date_of_upload
Played_history
PK,FK
Song_id
PK,FK2
User_id
PK
date_and_time
Payment_history
PK
Payment_id
paid_amount
PK,FK1
user-id
Payment_date
Mode_of_payment
Playlist_songs
PK,FK1
Playlist_id
PK,FK2
Song_id
Playlist
PK
Playlist_id
FK1
Creator_id
Playlist_name

Normalization

USER :-

User has **User_id** as its primary key but all values in the user table are not atomic as mob_no is a multivalued attribute.
Hence the User table is **not in 1NF.**

To convert it into 1NF, We should create another table containing mobile numbers for each user having User_id,mobile_num as primary key in that table.

Now user has (User_id,User_name,User_selected_lang,User_age, state, Nation)
User_id -> (User_name,User_selected_lang,User_age, state, Nation) can derive all attributes from the new table also all entries are atomic hence now in 1NF.
No Partial dependencies exist hence in 2NF.
No Transitive dependencies exist hence in 3NF.
All the attributes only depend on User_id (Primary key) and hence in **BCNF**.

Contact_details table (New table):
(User_id(FK,PK), mobile_no.(PK)) where User_id and mob_no are primary keys. As both attributes are part of the key itself hence in **BCNF**.

Artist :-

Artist has **Artist_id** as its primary key which identifies all other attributes.
All entries in the Artist table are atomic hence relation is in 1NF.
No Partial dependencies exist hence in 2NF.
No Transitive dependencies exist hence in 3NF.
All the attributes only depend on Artist_id (Primary key) and hence in **BCNF**.

Song :-

Table Song has Song_id as its primary key which identifies all other attributes.
All entries in the song table are atomic hence the relation is in 1NF.
No prime attributes derive any other non-prime attribute hence in 2NF.No Transitive dependencies exist hence in 3NF.
All the attributes only depend on Song_id (Primary key) and hence in **BCNF**.

Artist_type :-

Primary Key : Type_id.
All entries are atomic as no multivalued attributes present in the table, hence the relation is in 1NF.

No Partial dependencies exist hence in 2NF.

No Transitive dependencies exist hence in 3NF.

Type_name depends on Type_id (Primary key) and hence in **BCNF**.

Language :-

Primary Key : Language_id.
All entries are atomic as no multivalued attributes present in the table, hence the relation is in 1NF.

No Partial dependencies exist hence in 2NF.

No Transitive dependencies exist hence in 3NF.

Language_name depends on Language_id (Primary key) and hence in **BCNF**.

Genre :-

Primary Key : Genre_id.
All entries are atomic as no multivalued attributes present in the Genre table, hence the relation is in 1NF.

No Partial dependencies exist hence in 2NF.

No Transitive dependencies exist hence in 3NF.

Genre_name depends only on Genre_id (Primary key) and hence in **BCNF**.

Playlist :-

Primary Key : Playlist_id.
All entries are atomic as no multivalued attributes present in the table, hence the relation is in 1NF.

No Partial dependencies exist as both creator_id and playlist_name can be identified by playlist_id hence in 2NF.

No Transitive dependencies exist hence in 3NF.

Playlist_name and Creator_id depend on Playlist_id (Primary key) and hence in **BCNF**.

Similarly,

Liked_playlist , **Playlist_songs, Liked_song** all these tables don't have any multivalued attribute and attributes in respective tables are part of the primary key itself in their tables hence the tables are in **BCNF.**

Singer :-

Primary Key : Artist_id.
All entries are atomic as no multivalued attributes present in the table, hence the relation is in 1NF.
No Partial dependencies exist hence in 2NF.
No Transitive dependencies exist hence in 3NF.
Vocal_type and Singing_type depend on Artist_id (Primary key) and hence in **BCNF**.

Similarly, **Lyricist** and **Music_composer** both are in **BCNF**.

Customer_services :-

Primary Key : Service_id,User_id
Both primary as well as (Date_of_service,problem) are single valued attributes hence in 1NF.
No prime attributes derive any other non-prime attribute hence in 2NF.No Transitive dependencies exist hence in 3NF.
All attributes depend only on Primary key and hence in **BCNF**.

Played history :-

Primary Key : Song_id,User_id,date&time.
As a user can listen to a song multiple times a day that's why just song_id and user together cannot uniquely identify date&time.

There are only 3 attributes in the table and all 3 are part of the key hence the table is in **BCNF**.

Subscription_plan :-

Subscription_plan_id uniquely identifies the plan available in the system and no other attribute required, hence subscription_plan_id is the key in this table.
All other attributes (Plan_duration, Plan_amount, Plan_type) directly depend upon subscription_plan_id, Hence the table is in **BCNF**.

User_Subscribed_plan :-

Primary key consist of user_id,subscription_id and start_date as A user can take the same subscription after some days (month) hence to uniquely identify them
User_id+subscription_plan+Start_date is our primary key for this table.

All values are atomic hence it is in 1NF.
End_date depends on subscrption_id and start_date. Partial dependency exists, hence it is **not in 2NF**.

To remove this descripency we will we divide this into two

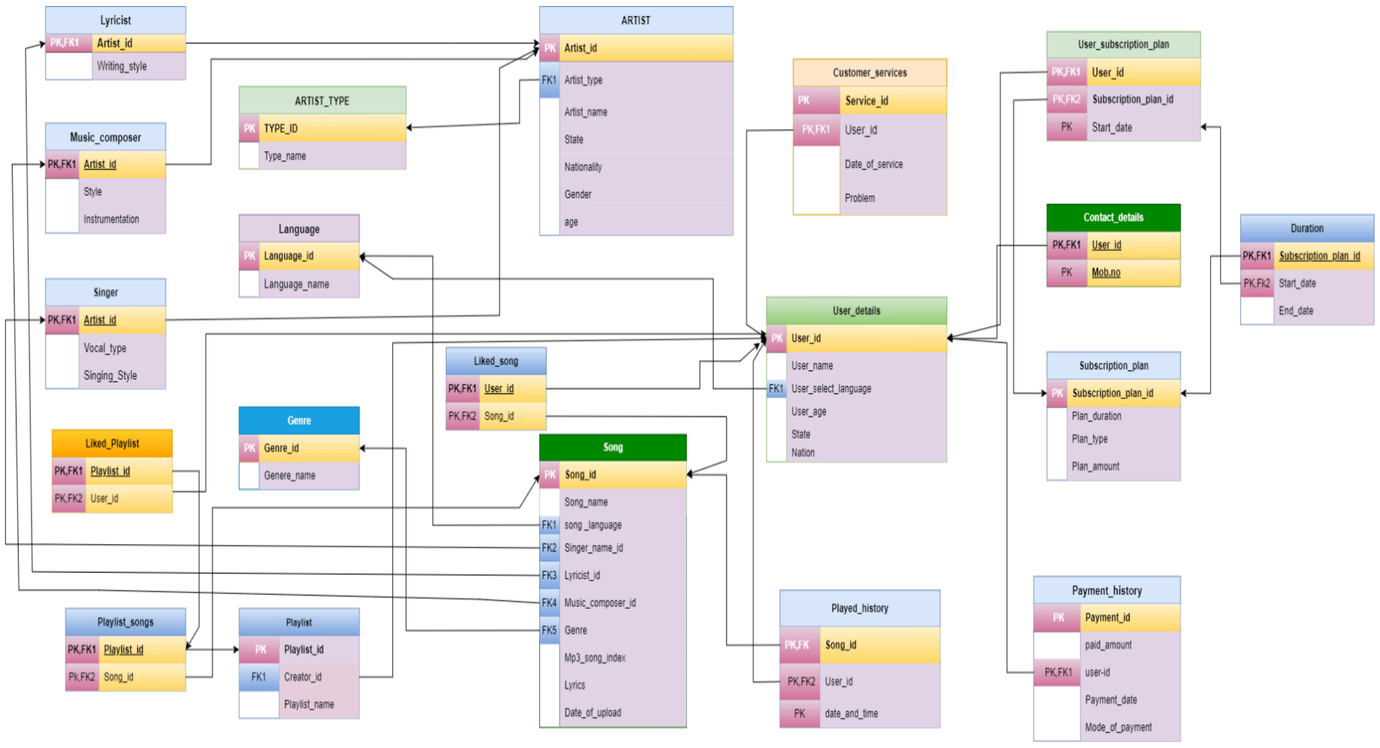
First table : R (user_id,subscription_id,start_date).
All are part of the primary key and all values are atomic; furthermore, no partial dependencies and no transitive dependencies hence table is in **BCNF** now.

Second table : R (subscription_id,start_date,end_date)
In this table subscription_id and start_date form the primary key , which uniquely identifies end_date.All values are atomic and no partial and transitive dependencies exist furthermore non prime i.e. end_date depends completely on whole primary key, hence this table is also in **BCNF** now.

Payment_history :-

Primary key : payment_id + User_id.
All entries are atomic as no multivalued attributes present in the table, hence the relation is in 1NF.No Partial dependencies exist hence in 2NF.No Transitive dependencies exist hence in 3NF.
All attributes depend only on Primary key, hence in **BCNF**.

ER schema after Normalization



Creation of table

```
Create Table Artist_Type(
   Type_Id Int Primary Key,
   Type_Name Varchar(25)
   );

   Create Table Language(
   Language_Id Int Primary Key,
   Language_Name Varchar(25)
   );


   Create Table Artist(
   Artist_Id Int Primary Key,
   Artist_Type Int,
   Artist_Name Varchar(25),
   State1 Varchar(25) Not Null,
   Nationality Varchar(25) Not Null,
   Gender Varchar(1) Not Null,
   Age Int Not Null,
   Foreign Key(Artist_Type) References Artist_Type(Type_Id)
   );

   Create Table Lyricist (
      Writing_Style Varchar(100),
      Artist_Id Int Primary Key,
      Foreign Key (Artist_Id) References Artist(Artist_Id)
   );

   Create Table Music_Composer (
      Artist_Id Int Primary Key,
      Music_Style Varchar(100),
    Instrumentation Varchar(100),
    Foreign Key (Artist_Id) References Artist(Artist_Id)
);

Create Table Singer (
    Artist_Id Int Primary Key,
    Vocal_Type Varchar(100),
    Singing_Style Varchar(100),
    Foreign Key (Artist_Id) References Artist(Artist_Id)
);

Create Table Genre(
Genre_Id Int Primary Key,
Genre_Name Varchar(25) Not Null
);

Create Table Song (
    Song_Id Int Primary Key,
    Song_Name Varchar(25),
    Song_Language Int,
    Singer_Id Int,
    Lyricist_Id Int,
    Music_Composer_Id Int,
    Genre Int,
    Mp3_Song_Index Varchar(30),
    -- this will be index to original file where the actual song will be stored
    Lyrics Clob,
    Date_Of_Upload Timestamp,
    Foreign Key (Song_Language) References Language(Language_Id),
    Foreign Key (Singer_Id) References Singer(Artist_Id),
    Foreign Key (Lyricist_Id) References Lyricist(Artist_Id),
    Foreign Key (Music_Composer_Id) References Music_Composer(Artist_Id),
    Foreign Key (Genre) References Genre(Genre_Id)
);


Create Table User_Details(
    User_Id Int Primary Key,
    User_Name Varchar(100),
    User_Selected_Language Int,
    User_Age Int Not Null,
    State Varchar(100),
    Nation Varchar(100),
    Foreign Key (User_Selected_Language) References Language(Language_Id)
);


Create Table Playlist (
    Playlist_Id Int Primary Key,
    Creator_Id Int,
    Playlist_Name Varchar(100),
    Foreign Key (Creator_Id) References User_Details(User_Id)
);

Create Table Playlist_Songs (
    Playlist_Id Int,
    Song_Id Int,
    Primary Key (Playlist_Id,Song_Id),
    Foreign Key (Playlist_Id) References Playlist(Playlist_Id),
    Foreign Key (Song_Id) References Song(Song_Id)
);

Create Table Liked_Song (
    User_Id Int,
    Song_Id Int,
    Primary Key (User_Id, Song_Id),
    Foreign Key (User_Id) References User_Details(User_Id),
    Foreign Key (Song_Id) References Song(Song_Id)
);

Create Table Liked_Playlist (
    Playlist_Id Int,
    User_Id Int ,
    Primary Key (Playlist_Id,User_Id),
      Foreign Key (Playlist_Id) References Playlist(Playlist_Id),
      Foreign Key (User_Id) References User_Details(User_Id)
   );

   Create Table Played_History (
      Song_Id Int,
      User_Id Int,
      Date_And_Time Timestamp,
      Primary Key (Song_Id,User_Id,Date_And_Time),
      Foreign Key (Song_Id) References Song(Song_Id),
      Foreign Key (User_Id) References User_Details(User_Id)
   );

   Create Table Subscription_Plan (
      Subscription_Plan_Id Int Primary Key,
      Plan_Duration Int,--in months
      Plan_Type Varchar(25),
      Plan_Amount Decimal(6,2)
   );

   Create Table User_Subscription_Plan (
      User_Id Int,
      Subscription_Plan_Id Int,
      Start_Date Date ,
      Primary Key (User_Id,Subscription_Plan_Id,Start_Date),
      Foreign Key (User_Id) References User_Details(User_Id),
      Foreign Key (Subscription_Plan_Id) References Subscription_Plan
(Subscription_Plan_Id)
   );

   Create Table Payment_History (
      Payment_Id Int,
      Paid_Amount Decimal(10, 2),
      User_Id Int,
      Payment_Date Date,
      Mode_Of_Payment Varchar(100),
      Primary Key (Payment_Id,User_Id),
      Foreign Key (User_Id) References User_Details(User_Id)
   );

   Create Table Customer_Services (
      Service_Id Int,
      User_Id Int,
      Date_Of_Service Date,
      Problem Varchar(100),
      Primary Key (Service_Id,User_Id),
      Foreign Key (User_Id) References User_Details(User_Id)
   );

   Create Table Contact_Details(
      User_Id Int,
      Mob_No Number(10) Not Null,
      Primary Key (User_Id,Mob_No),
      Foreign Key (User_Id) References User_Details(User_Id)
   );

   Create Table Duration(
      Subscription_Plan_Id Int,
      Start_Date Date,
      End_Date Date,
      Primary Key (Subscription_Plan_Id,Start_Date),
      Foreign Key (Subscription_Plan_Id) References Subscription_Plan(
Subscription_Plan_Id)
   );
```

DATA INSERTION

--- ARTIST TYPE

```
INSERT INTO ARTIST_TYPE VALUES(1,'SINGER');
INSERT INTO ARTIST_TYPE VALUES(2,'MUSIC_COMPOSER');
INSERT INTO ARTIST_TYPE VALUES(3,'LYRICIST');

SELECT * FROM ARTIST_TYPE;
DESC ARTIST_TYPE;
```

--- GENRE

```
INSERT INTO GENRE VALUES (1,'CLASSICAL MUSIC');
INSERT INTO GENRE VALUES (2,'INDIAN FOLK');
INSERT INTO GENRE VALUES (3,'INDIE');
INSERT INTO GENRE VALUES (4,'ROMANTIC');
INSERT INTO GENRE VALUES (5,'PARTY MUSIC');
INSERT INTO GENRE VALUES (6,'WESTERN');
INSERT INTO GENRE VALUES (7,'BOLLYWOOD');
INSERT INTO GENRE VALUES (8,'TOLLYWOOD MUSIC');
INSERT INTO GENRE VALUES (9,'POPULAR MUSIC');

SELECT * FROM GENRE;
DESC GENRE;
```

---- LANGUAGES

```
   INSERT INTO LANGUAGE VALUES (1,'ENGLISH');
   INSERT INTO LANGUAGE VALUES (2,'TELGU');
   INSERT INTO LANGUAGE VALUES (3,'HINDI');
   INSERT INTO LANGUAGE VALUES (4,'MARATHI');
   INSERT INTO LANGUAGE VALUES (5,'RAJASTHANI');
   INSERT INTO LANGUAGE VALUES (6,'PUNJABI');
   INSERT INTO LANGUAGE VALUES (7,'TAMIL');
   INSERT INTO LANGUAGE VALUES (8,'HARYANVI');
   INSERT INTO LANGUAGE VALUES (9,'URDU');
   INSERT INTO LANGUAGE VALUES (10,'BENGALI');


   SELECT * FROM LANGUAGE;
   DESC LANGUAGE;
```

--- ARTIST

```
   INSERT INTO ARTIST VALUES(1,1,'ARIJIT SINGH','WEST
BENGAL','INDIA','M',38);
   INSERT INTO ARTIST VALUES(2,2,'ANIRUDH RAVICHANDER','TAMIL
NADU','INDIA','M',35);
   INSERT INTO ARTIST VALUES(3,1,'DEVI SHREE PRASAD','ANDHRA
PRADESH','INDIA','M',44);
   INSERT INTO ARTIST VALUES(4,1,'NEHA
KAKKAR','UTTARAKHAND','INDIA','F',35);
   INSERT INTO ARTIST VALUES (5,1,'LATA MANGESHKAR','MADHYA
PRADESH','INDIA','F',92);
   INSERT INTO ARTIST VALUES (6,3,'PRADEEP
KUMAR','MAHARAHSTRA','INDIA','M',82);
   INSERT INTO ARTIST VALUES (7,2,'C.
RAMCHANDRA','MAHARAHSTRA','INDIA','M',63);
```

```
   INSERT INTO ARTIST VALUES (8,2,'AJAY ATUL','MAHARAHSTRA','INDIA','M',30);
   INSERT INTO ARTIST VALUES (9,2,'A.R.REHMAN','GUJRAT','INDIA','M',43);
   INSERT INTO ARTIST VALUES (10,3,'GULZAR','PUNJAB','INDIA','M',60);
   INSERT INTO ARTIST VALUES (11,3,'AMITABH
BHATACHARYA','UP','INDIA','M',35);
   INSERT INTO ARTIST VALUES (12,1,'TAYLOR
SWIFT','PENNSYLVANIA','US','F',34);
   INSERT INTO ARTIST VALUES (13,1,'VIKRAM','PUNJAB','PAKISTHAN','M',20);
   INSERT INTO ARTIST VALUES (14,2,'SHEKHAR','YORKSHIRE','ENGLAND','M',25);
   INSERT INTO ARTIST VALUES (15,3,'HERAMB','HAWAII','US','M',30);


   SELECT * FROM ARTIST;
   DESC ARTIST;
```

--- LYRICIST

```
   INSERT INTO LYRICIST VALUES ('ARTISTIC WRITING SONG AND POEM',6);
   INSERT INTO LYRICIST VALUES ('JAZZ AND ROCK MUSIC LYRICS',15);
   INSERT INTO LYRICIST VALUES ('FILM MUSIC LYRICS',11);
   INSERT INTO LYRICIST VALUES ('POETRY AND DEEP MEANING LYRICS',10);


   SELECT * FROM LYRICIST;
   DESC LYRICIST;
```

---SINGER

```
   INSERT INTO SINGER VALUES (1,'SOPRANO','BOLLYWOOD HINDI MUSIC AND
CALM SONGS');
   INSERT INTO SINGER VALUES (3,'TENOR','FILM MUSIC');
   INSERT INTO SINGER VALUES (4,'CONTRALTO','BOLLYWOOD HINDI MUSIC
AND CALM SONGS');
   INSERT INTO SINGER VALUES (5,'SOPRANO','ALL TYPE OF MUSIC');
   INSERT INTO SINGER VALUES (13,'TENOR','LOUD AND ROCK');
   INSERT INTO SINGER VALUES (12,'CONTRALTO','SINGS INDIE SONGS...
FAMOUS WORLDWIDE');


   SELECT * FROM SINGER;
   DESC SINGER;
```

---- MUSIC COMPOSER

```
   INSERT INTO MUSIC_COMPOSER VALUES (2,'HIGH INSTRUMENTAL AND FILM
MUSIC','ALL');
   INSERT INTO MUSIC_COMPOSER VALUES (7,'FILM MUSIC','ALL');
   INSERT INTO MUSIC_COMPOSER VALUES (9,'FILM MUSIC , CLASSICAL AND
MODERN MUSIC','ALL');
   INSERT INTO MUSIC_COMPOSER VALUES (8,'FILM MUSIC MARATHI AND
OTHER LANG MUSIC','ALL');
   INSERT INTO MUSIC_COMPOSER VALUES (14,'INDIE COMPOSER','ALL');

   SELECT * FROM MUSIC_COMPOSER;
   DESC MUSIC_COMPOSER;
```

---- SONG

```
   INSERT INTO SONG VALUES(1,'AE MERE WATTAN KE LOGO',3,5,6,7,1,'1','AYE
MERE VATAN KE LOGON,ZARA AANKH MEIN BHAR LO PAANI,JO SHAHEED HUE
HAIN UNKI,ZARA YAAD KARO QURBAANI',TO_TIMESTAMP('1963-01-26 12:30:45',
'YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Song VALUES (2, 'Channa Mereya', 3, 3, 6, 7, 2, 'index2.mp3', 'Lyrics
for Channa Mereya...', TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(3, 'Kal Ho Naa Ho', 3, 4, 11, 9, 4, 'index3.mp3', 'Lyrics
for Kal Ho Naa Ho...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(4, 'Tere Bina', 3, 5, 10, 8, 1, 'index4.mp3', 'Lyrics for
Tere Bina...', TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(5, 'Tera Ban Jaunga', 3, 13, 15, 14, 1, 'index5.mp3',
'Lyrics for Tera Ban Jaunga...', TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(6, 'Jeene Laga Hoon', 3, 1, 10,8, 1 ,'index6.mp3',
'Lyrics for Jeene Laga Hoon...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES (7, 'Raabta', 3, 4, 15, 2, 1 ,'index7.mp3', 'Lyrics for
Raabta...', TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(8, 'Tera Hone Laga Hoon', 3, 1, 10,8, 7, 'index8.mp3',
'Lyrics for Tera Hone Laga Hoon...', TO_TIMESTAMP('2024-02-28','YYYY-MM-DD'));
   INSERT INTO Song VALUES(9, 'Tum Se Hi', 3, 5, 11, 2, 9, 'index9.mp3', 'Lyrics for
Tum Se Hi...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(10, 'Agar Tum Saath Ho', 3, 1, 15, 8, 1, 'index10.mp3',
'Lyrics for Tera Ghata...', TO_TIMESTAMP('2024-02-13','YYYY-MM-DD'));
   INSERT INTO Song VALUES(11, 'Tum Mile', 3, 3, 15, 7, 2, 'index11.mp3', 'Lyrics for
Tum Mile...', TO_TIMESTAMP('2023-02-08','YYYY-MM-DD'));
   INSERT INTO Song VALUES(12, 'Tum Jo Aaye', 3, 12, 11, 14, 3, 'index12.mp3',
'Lyrics for Tum Jo Aaye...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(13, 'Tera Yaar Hoon Main', 3, 12, 15, 7, 4,
'index13.mp3', 'Lyrics for Tera Yaar Hoon Main...',
TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(14, 'Tum Se Hi', 3, 4, 11, 7, 5, 'index14.mp3', 'Lyrics
for Tum Se Hi...', TO_TIMESTAMP('1998-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(15, 'Tera Zikr', 3, 5, 11, 14, 6, 'index15.mp3', 'Lyrics for
Tera Zikr...', TO_TIMESTAMP('2000-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(16, 'Flowers', 1, 1, 6, 2, 1, 'index16.mp3', 'Lyrics for
Flowers...', TO_TIMESTAMP('2002-01-30','YYYY-MM-DD'));
```

```
   INSERT INTO Song VALUES(17, 'Popular', 1, 3, 15, 7, 2, 'index17.mp3', 'Lyrics for
Popular...', TO_TIMESTAMP('2008-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(18, 'One Of The Girls', 1, 12, 15, 9, 4, 'index18.mp3',
'Lyrics for One Of The Girls...', TO_TIMESTAMP('2010-02-02','YYYY-MM-DD'));
   INSERT INTO Song VALUES(19, 'The Nights', 1, 5, 10, 8, 5, 'index19.mp3', 'Lyrics
for Love Like This...', TO_TIMESTAMP('2007-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(20, 'Paint The Town Red', 1, 13, 15, 9, 6,
'index20.mp3', 'Lyrics for Paint The Town Red...',
TO_TIMESTAMP('2006-08-18','YYYY-MM-DD'));
   INSERT INTO Song VALUES(21, 'Enchanted ', 1, 12, 11, 14, 3, 'index21.mp3', 'Lyrics
for Enchanted...', TO_TIMESTAMP('2002-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(22, 'Kannaa Nidurinchara', 2, 1, 6, 2, 1,
'index22.mp3', 'Telugu song1 lyrics...', TO_TIMESTAMP('2023-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(23, 'Telugusong2', 2, 3, 6, 7, 2, 'index23.mp3', 'Telugu
song2 lyrics...', TO_TIMESTAMP('2022-12-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(24, 'Telugusong3', 2, 4, 11, 9, 4, 'index24.mp3',
'Telugu song3 lyrics...', TO_TIMESTAMP('2020-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(25, 'Telugusong4', 2, 5, 10, 8, 5, 'index25.mp3',
'Telugu song4 lyrics...', TO_TIMESTAMP('2022-04-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(26, 'Telugusong5', 2, 13, 15, 14, 6, 'index26.mp3',
'Telugu song5 lyrics...', TO_TIMESTAMP('2021-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(27, 'Telugusong6', 2, 3, 15, 7, 7, 'index27.mp3', 'Telugu
song6 lyrics...', TO_TIMESTAMP('2020-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(28, 'Rajasthani song1', 5, 1, 6, 2, 1, 'index28.mp3',
'Rajasthani song1 lyrics...', TO_TIMESTAMP('2023-04-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(29, 'Rajasthani song2', 5, 3, 6, 7, 2, 'index29.mp3',
'Rajasthani song2 lyrics...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(30, 'ZINGAAT', 4, 1, 6, 2, 1, 'index30.mp3', 'Marathi
song1 lyrics...', TO_TIMESTAMP('2024-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(31, 'Tum Hi Ho', 3, 1, 6, 2, 1, 'index1.mp3', 'Lyrics for
Tum Hi Ho...', TO_TIMESTAMP('2024-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(32, 'COCKTAIL', 3, 1, 6, 2, 1, 'index1.mp3', 'Lyrics for
COCKTAIL...', TO_TIMESTAMP('2018-01-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(33, 'MAIN KOI AISA GEET GAAO', 3, 1, 6, 2, 1,
'index1.mp3', 'Lyrics for MAIN KOI ...',
TO_TIMESTAMP('1992-03-30','YYYY-MM-DD'));
```

```
   INSERT INTO Song VALUES(34, 'AJAB SI', 3, 1, 6, 2, 1, 'index1.mp3', 'AJAB SI',
TO_TIMESTAMP('2004-03-30','YYYY-MM-DD'));
   INSERT INTO Song VALUES(35, 'SATHIYA', 3, 1, 6, 2, 1, 'index1.mp3', 'Lyrics
SATHIYA...', TO_TIMESTAMP('2006-03-30','YYYY-MM-DD'));

   SELECT * FROM SONG;
   DESC SONG;
```

--- USER DETAILS

```
   INSERT INTO USER_DETAILS VALUES
(101,'KARTIK',1,20,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(102,'SHEKHAR',5,24,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES (103,'HERAMB',2,18,'ANDRA
PRADESH','INDIA');
   INSERT INTO USER_DETAILS VALUES (104,'TEJA',2,22,'TELANGANA','INDIA');
   INSERT INTO USER_DETAILS VALUES (105,'PRAHANT',3,35,'BIHAR','INDIA');
   INSERT INTO USER_DETAILS VALUES
(106,'SUYASH',4,42,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(107,'DEVENDRA',4,26,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(108,'JIGNESH',6,22,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(109,'PRATHAMESH',1,50,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(110,'GANYA',4,32,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(111,'VIGHNESH',4,29,'MAHARASHTRA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(112,'ANUBHAV',3,25,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES
(113,'ANIKET',3,24,'MAHARASHTRA','INDIA');
```

```
   INSERT INTO USER_DETAILS VALUES (114,'TANMAY',5,16,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES (115,'GIRISH',1,33,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES (116,'KPK',1,26,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES (117,'SONI',5,25,'RAJSTHAN','INDIA');
   INSERT INTO USER_DETAILS VALUES
(118,'NIKESH',2,34,'TELANGANA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(119,'SAICHARAN',2,22,'TELANGANA','INDIA');
   INSERT INTO USER_DETAILS VALUES (120,'VINAY',2,24,'TELANGANA','INDIA');
   INSERT INTO USER_DETAILS VALUES (121,'DHANUSH',2,18,'ANDRA
PRADESH','INDIA');
   INSERT INTO USER_DETAILS VALUES
(122,'VAISHANAVI',1,19,'KARNATAKA','INDIA');
   INSERT INTO USER_DETAILS VALUES (123,'NEHA',2,17,'BIHAR','INDIA');
   INSERT INTO USER_DETAILS VALUES (124,'KIRTI',3,28,'BIHAR','INDIA');
   INSERT INTO USER_DETAILS VALUES (125,'MONYA',2,32,'BIHAR','INDIA');
   INSERT INTO USER_DETAILS VALUES
(126,'SHRAVANI',4,19,'KARNATAKA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(127,'AKSHAY',3,12,'KARNATAKA','INDIA');
   INSERT INTO USER_DETAILS VALUES
(128,'SALMAN',2,19,'KARNATAKA','INDIA');
   INSERT INTO USER_DETAILS VALUES (129,'ALEXA',1,19,'CALIFORNIA','US');
   INSERT INTO USER_DETAILS VALUES (130,'JOHN',1,29,'LOS ANJELES','US');
   INSERT INTO USER_DETAILS VALUES (131,'A B
DEVILLIERS',1,25,'PRETORIA','SOUTH AFRICA');
   INSERT INTO USER_DETAILS VALUES
(132,'MERRY',3,35,'LAHORE','PAKISTHAN');
   INSERT INTO USER_DETAILS VALUES (133,'John
Smith',1,45,'TORONTO','CANADA');
   INSERT INTO USER_DETAILS VALUES (134,'Emma
Johnson',1,25,'VANCOUVER','CANADA');
   INSERT INTO USER_DETAILS VALUES (135,'Michael
Brown',1,19,'SYDNEY','AUSTRALIA');
   INSERT INTO USER_DETAILS VALUES (136,'Sophia
Martin',1,20,'PARIS','FRANCE');
```

```
   INSERT INTO USER_DETAILS VALUES (137,'William
Garcia',1,50,'BERLIN','GERMANY');
   INSERT INTO USER_DETAILS VALUES (138,'Olivia
Martinez',1,17,'MUNICH','FRANCE');
   INSERT INTO USER_DETAILS VALUES (139,'James Taylor',1,35,'LYON','FRANCE');
   INSERT INTO USER_DETAILS VALUES (140,'VISHAL',4,25,'LYON','FRANCE');

   SELECT * FROM USER_DETAILS;
   DESC USER_DETAILS;
```

---- played history

```
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (17,
115, TO_TIMESTAMP('2024-03-15 09:23:15','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (2,
112, TO_TIMESTAMP('2024-03-03 14:45:32','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
118, TO_TIMESTAMP('2024-03-11 18:12:50','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (22,
128, TO_TIMESTAMP('2024-03-01 22:37:04','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (1,
120, TO_TIMESTAMP('2024-03-25 08:59:21','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 127, TO_TIMESTAMP('2024-02-21 16:28:39','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (32,
116, TO_TIMESTAMP('2024-02-29 11:07:55','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (1,
106, TO_TIMESTAMP('2024-02-25 19:34:17','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (1,
109, TO_TIMESTAMP('2024-03-10 13:49:28','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (33,
115, TO_TIMESTAMP('2024-02-19 20:05:36','YYYY-MM-DD HH24:MI:SS'));
```

```
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (18,
101, TO_TIMESTAMP('2024-02-24 09:17:42','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (22,
116, TO_TIMESTAMP('2024-03-05 15:56:58','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (15,
125, TO_TIMESTAMP('2024-02-28 23:28:10','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(30, 107, TO_TIMESTAMP('2024-03-12 07:33:27','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (19,
129, TO_TIMESTAMP('2024-02-17 17:09:43','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (13,
117, TO_TIMESTAMP('2024-03-21 10:25:51','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (19,
123, TO_TIMESTAMP('2024-03-03 20:47:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (31,
108, TO_TIMESTAMP('2024-02-14 12:16:19','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (19,
112, TO_TIMESTAMP('2024-02-29 06:55:34','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 129, TO_TIMESTAMP('2024-02-17 18:23:48','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 113, TO_TIMESTAMP('2024-02-21 09:41:05','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
131, TO_TIMESTAMP('2024-02-14 21:12:17','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (2,
111, TO_TIMESTAMP('2024-03-03 13:38:32','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (19,
130, TO_TIMESTAMP('2024-02-19 16:57:49','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (16,
110, TO_TIMESTAMP('2024-03-10 10:02:05','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(20, 125, TO_TIMESTAMP('2024-03-01 17:08:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (21,
125, TO_TIMESTAMP('2024-03-06 21:08:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 121, TO_TIMESTAMP('2024-03-08 05:44:20','YYYY-MM-DD HH24:MI:SS'));
```

```
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (2,
119, TO_TIMESTAMP('2024-02-25 14:51:52','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (22,
130, TO_TIMESTAMP('2024-03-01 19:39:21','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 129, TO_TIMESTAMP('2024-03-19 09:36:48','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
129, TO_TIMESTAMP('2018-07-15 09:23:15','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(20, 127, TO_TIMESTAMP('2019-09-03 14:45:32','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (5,
125, TO_TIMESTAMP('2020-05-11 18:12:50','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(30, 123, TO_TIMESTAMP('2022-08-01 22:37:04','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 124, TO_TIMESTAMP('2021-11-25 08:59:21','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (17,
130, TO_TIMESTAMP('2020-03-21 16:28:39','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(30, 113, TO_TIMESTAMP('2019-02-29 11:07:55','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (22,
131, TO_TIMESTAMP('2019-12-25 19:34:17','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (12,
126, TO_TIMESTAMP('2023-03-10 13:49:28','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
131, TO_TIMESTAMP('2023-04-19 20:05:36','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (7,
127, TO_TIMESTAMP('2018-06-24 09:17:42','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (17,
129, TO_TIMESTAMP('2021-01-05 15:56:58','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (1,
128, TO_TIMESTAMP('2019-07-28 23:28:10','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (16,
124, TO_TIMESTAMP('2018-10-12 07:33:27','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(14, 130, TO_TIMESTAMP('2022-04-17 17:09:43','YYYY-MM-DD HH24:MI:SS'));
```

```
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (13,
122, TO_TIMESTAMP('2021-02-21 10:25:51','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (9,
132, TO_TIMESTAMP('2019-03-03 20:47:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
124, TO_TIMESTAMP('2023-06-14 12:16:19','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (7,
114, TO_TIMESTAMP('2018-12-29 06:55:34','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (15,
116, TO_TIMESTAMP('2020-07-17 18:23:48','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (11,
129, TO_TIMESTAMP('2019-11-21 09:41:05','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (17,
123, TO_TIMESTAMP('2020-02-14 21:12:17','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
112, TO_TIMESTAMP('2021-08-19 16:57:49','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (22,
128, TO_TIMESTAMP('2018-04-10 10:02:05','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (4,
130, TO_TIMESTAMP('2019-11-01 17:08:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (18,
126, TO_TIMESTAMP('2022-03-06 21:08:03','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (3,
125, TO_TIMESTAMP('2018-08-10 18:40:17','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (17,
131, TO_TIMESTAMP('2021-05-11 13:56:29','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES (19,
118, TO_TIMESTAMP('2020-12-03 14:34:28','YYYY-MM-DD HH24:MI:SS'));
   INSERT INTO Played_History (Song_Id, User_Id, Date_And_Time) VALUES
(10, 115, TO_TIMESTAMP('2019-08-22 05:57:40','YYYY-MM-DD HH24:MI:SS'));

   select * from Played_History;
   DESC Played_History;
```

------ SUBSCRIPTION PLAN

```
   INSERT INTO SUBSCRIPTION_PLAN VALUES(1, 12, 'SILVER', 99.99);
   INSERT INTO SUBSCRIPTION_PLAN VALUES (2, 6, 'GOLD', 149.99);
   INSERT INTO SUBSCRIPTION_PLAN VALUES (3, 3, 'PLATINUM', 199.99);
   INSERT INTO SUBSCRIPTION_PLAN VALUES (4, 1, 'BASIC', 49.99);
   INSERT INTO SUBSCRIPTION_PLAN VALUES (5, 24, 'PREMIUM', 299.99);

   select * from SUBSCRIPTION_PLAN;
   DESC SUBSCRIPTION_PLAN;
```

---- USER SUBSCRIBED PLANS

```
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (101, 1,
TO_DATE('2024-03-29','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (105, 2,
TO_DATE('2024-03-29','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (119, 3,
TO_DATE('2024-03-29','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (128, 3,
TO_DATE('2022-02-23','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (111, 1,
TO_DATE('2024-01-29','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (111, 4,
TO_DATE('2023-12-13','YYYY-MM-DD'));
   INSERT INTO USER_SUBSCRIPTION_PLAN VALUES (112, 5,
TO_DATE('2024-01-24','YYYY-MM-DD'));

   SELECT * FROM USER_SUBSCRIPTION_PLAN;
   DESC USER_SUBSCRIPTION_PLAN;
```

--playlist

```
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (1, 101,
'Chill Vibes');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (2, 106,
'Workout Jams');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (3, 123,
'desi bollywood');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (4, 121,
'geet');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (5, 123,
'rock songs');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (6, 123,
'party mood');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (7, 102,
'all at once');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (8, 101,
'English popular');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (9, 111,
'indian indie');
   INSERT INTO Playlist (Playlist_id, Creator_id, Playlist_name) VALUES (10, 122,
'poetic');

   SELECT * FROM PLAYLIST;
   DESC Playlist;
```

--- playlist songs

```
   INSERT INTO playlist_songs VALUES (1,2);INSERT INTO playlist_songs VALUES
(8,19);
   INSERT INTO playlist_songs VALUES (1,11);INSERT INTO playlist_songs
VALUES (8,8);
   INSERT INTO playlist_songs VALUES (1,12);INSERT INTO playlist_songs
VALUES (8,15);
```

```
    INSERT INTO playlist_songs VALUES (1,21);INSERT INTO playlist_songs
VALUES (8,25);
    INSERT INTO playlist_songs VALUES (1,27);INSERT INTO playlist_songs
VALUES (8,31);
    INSERT INTO playlist_songs VALUES (2,21);INSERT INTO playlist_songs
VALUES (9,10);
    INSERT INTO playlist_songs VALUES (2,12);INSERT INTO playlist_songs
VALUES (9,11);
    INSERT INTO playlist_songs VALUES (3,5);INSERT INTO playlist_songs VALUES
(9,12);
    INSERT INTO playlist_songs VALUES (3,6);INSERT INTO playlist_songs VALUES
(9,13);
    INSERT INTO playlist_songs VALUES (3,8);INSERT INTO playlist_songs VALUES
(9,14);
    INSERT INTO playlist_songs VALUES (4,1);INSERT INTO playlist_songs VALUES
(9,17);
    INSERT INTO playlist_songs VALUES (4,9);INSERT INTO playlist_songs VALUES
(9,19);
    INSERT INTO playlist_songs VALUES (5,2);INSERT INTO playlist_songs VALUES
(10,10);
    INSERT INTO playlist_songs VALUES (5,9);INSERT INTO playlist_songs VALUES
(10,31);
    INSERT INTO playlist_songs VALUES (6,22);INSERT INTO playlist_songs
VALUES (10,29);
    INSERT INTO playlist_songs VALUES (6,17);INSERT INTO playlist_songs
VALUES (10,28);
    INSERT INTO playlist_songs VALUES (6,16);INSERT INTO playlist_songs
VALUES (7,20);
    INSERT INTO playlist_songs VALUES (6,8);INSERT INTO playlist_songs VALUES
(7,21);
    INSERT INTO playlist_songs VALUES (7,6);INSERT INTO playlist_songs VALUES
(7,23);
    INSERT INTO playlist_songs VALUES (7,10);INSERT INTO playlist_songs
VALUES (7,29);

    SELECT * FROM playlist_songs;
    DESC playlist_songs;
```

--- LIKED SONG;

```
   INSERT INTO liked_song (user_id, song_id) VALUES (101,1);INSERT INTO
liked_song (user_id, song_id) VALUES (102,1);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,13);INSERT INTO
liked_song (user_id, song_id) VALUES (102,12);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,17);INSERT INTO
liked_song (user_id, song_id) VALUES (106,19);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,20);INSERT INTO
liked_song (user_id, song_id) VALUES (131,16);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,3);INSERT INTO
liked_song (user_id, song_id) VALUES (123,14);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,26);INSERT INTO
liked_song (user_id, song_id) VALUES (115,18);
   INSERT INTO liked_song (user_id, song_id) VALUES (101,30);INSERT INTO
liked_song (user_id, song_id) VALUES (116,19);
   INSERT INTO liked_song (user_id, song_id) VALUES (118,12);INSERT INTO
liked_song (user_id, song_id) VALUES (121,18);
   INSERT INTO liked_song (user_id, song_id) VALUES (119,4);INSERT INTO
liked_song (user_id, song_id) VALUES (121,1);
   INSERT INTO liked_song (user_id, song_id) VALUES (120,31);INSERT INTO
liked_song (user_id, song_id) VALUES (110,17);
   INSERT INTO liked_song (user_id, song_id) VALUES (121,11);INSERT INTO
liked_song (user_id, song_id) VALUES (119,12);
   INSERT INTO liked_song (user_id, song_id) VALUES (113,13);INSERT INTO
liked_song (user_id, song_id) VALUES (117,1);
   INSERT INTO liked_song (user_id, song_id) VALUES (132, 11); INSERT INTO
liked_song (user_id, song_id) VALUES (123, 17);
   INSERT INTO liked_song (user_id, song_id) VALUES (111, 28); INSERT INTO
liked_song (user_id, song_id) VALUES (107, 24);
   INSERT INTO liked_song (user_id, song_id) VALUES (130, 31); INSERT INTO
liked_song (user_id, song_id) VALUES (117, 6);
   INSERT INTO liked_song (user_id, song_id) VALUES (129, 7); INSERT INTO
liked_song (user_id, song_id) VALUES (120, 9);
   INSERT INTO liked_song (user_id, song_id) VALUES (117, 1); INSERT INTO
liked_song (user_id, song_id) VALUES (128, 8);
```

INSERT INTO liked_song (user_id, song_id) VALUES (127, 15); INSERT INTO liked_song (user_id, song_id) VALUES (126, 23);
INSERT INTO liked_song (user_id, song_id) VALUES (125, 16); INSERT INTO liked_song (user_id, song_id) VALUES (104, 21);
INSERT INTO liked_song (user_id, song_id) VALUES (123, 20); INSERT INTO liked_song (user_id, song_id) VALUES (122, 11);
INSERT INTO liked_song (user_id, song_id) VALUES (119, 14); INSERT INTO liked_song (user_id, song_id) VALUES (104, 31);
INSERT INTO liked_song (user_id, song_id) VALUES (116, 10); INSERT INTO liked_song (user_id, song_id) VALUES (117, 31);

DESC liked_song;
SELECT * FROM liked_song;

--- LIKED PLAYLIST

INSERT INTO liked_playlist (playlist_id, user_id) VALUES (1, 102);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (2, 122);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (3, 114);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (4, 127);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (5, 102);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (6, 119);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (7, 111);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (8, 130);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (9, 105);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (10, 116);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (1, 125);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (2, 109);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (3, 118);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (4, 131);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (5, 113);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (6, 128);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (7, 120);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (8, 132);
INSERT INTO liked_playlist (playlist_id, user_id) VALUES (9, 115);

```
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (10, 126);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (5, 114);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (9, 130);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (7, 117);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (3, 126);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (8, 119);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (10, 132);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (6, 115);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (4, 128);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (2, 120);
    INSERT INTO liked_playlist (playlist_id, user_id) VALUES (1, 132);

    DESC liked_playlist;
    SELECT * FROM liked_playlist;
```

----PAYMENT HISTORY

```
    INSERT INTO payment_history VALUES (1, 150.75, 101 ,TO_DATE('2024-03-29
12:30:13','YYYY-MM-DD HH24:MI:SS'), 'Credit Card');
    INSERT INTO payment_history VALUES (2, 200, 131 ,TO_DATE('2022-04-20
18:30:13','YYYY-MM-DD HH24:MI:SS'), 'Cash');
    INSERT INTO payment_history VALUES (3, 350.75, 123 ,TO_DATE('2023-02-18
16:20:13','YYYY-MM-DD HH24:MI:SS'), 'PayPal');
    INSERT INTO payment_history VALUES (4, 400.75, 105 ,TO_DATE('2024-03-12
2:20:33','YYYY-MM-DD HH24:MI:SS'), 'Credit Card');
    INSERT INTO payment_history VALUES (5, 150, 108 ,TO_DATE('2022-05-10
8:32:43','YYYY-MM-DD HH24:MI:SS'), 'Bank Transfer');
    INSERT INTO payment_history VALUES (6, 300, 105 ,TO_DATE('2023-05-29
22:15:38','YYYY-MM-DD HH24:MI:SS'), 'Mobile Wallet');
    INSERT INTO payment_history VALUES (7, 75, 108 ,TO_DATE('2024-03-29
19:24:13','YYYY-MM-DD HH24:MI:SS'), 'PayPal');
    INSERT INTO payment_history VALUES (8, 725, 105 ,TO_DATE('2024-03-20
12:30:13','YYYY-MM-DD HH24:MI:SS'), 'Cash');
```

```
   SELECT * FROM PAYMENT_HISTORY;
   DESC PAYMENT_HISTORY;
```

---- CUSTOMER SERVICES

```
   INSERT INTO customer_services VALUES (1, 101 ,TO_DATE('2024-03-29
12:30:13','YYYY-MM-DD HH24:MI:SS'), 'Network connectivity issue');
   INSERT INTO customer_services VALUES (2, 121 ,TO_DATE('2024-03-12
2:20:33','YYYY-MM-DD HH24:MI:SS'), 'Data recovery request');
   INSERT INTO customer_services VALUES (3, 115 ,TO_DATE('2022-05-10
8:32:43','YYYY-MM-DD HH24:MI:SS'), 'Software installation problem');
   INSERT INTO customer_services VALUES (4, 121 ,TO_DATE('2024-03-20
12:30:13','YYYY-MM-DD HH24:MI:SS'), 'Data recovery request');
   INSERT INTO customer_services VALUES (5, 111 ,TO_DATE('2022-05-10
8:32:43','YYYY-MM-DD HH24:MI:SS'), 'Password reset assistance');
   INSERT INTO customer_services VALUES (6, 105 ,TO_DATE('2024-03-29
12:30:13','YYYY-MM-DD HH24:MI:SS'), 'Software installation problem');

   SELECT * FROM customer_services;
   desc customer_services;
```

--- CONTACT DETAILS

```
   INSERT INTO contact_details VALUES (101, 9876543210);
   INSERT INTO contact_details VALUES (101, 8872985628);
   INSERT INTO contact_details VALUES (102, 9462863284);
   INSERT INTO contact_details VALUES (103, 9767562894);
   INSERT INTO contact_details VALUES (104, 9834758757);
   INSERT INTO contact_details VALUES (104, 7876543210);
   INSERT INTO contact_details VALUES (104, 9782364870);
   INSERT INTO contact_details VALUES (105, 8543534230);
   INSERT INTO contact_details VALUES (106, 8525252450);
   INSERT INTO contact_details VALUES (106, 7352525250);
   INSERT INTO contact_details VALUES (107, 9325423520);

   SELECT * FROM contact_details;
   DESC contact_details;
```

--- DURATION

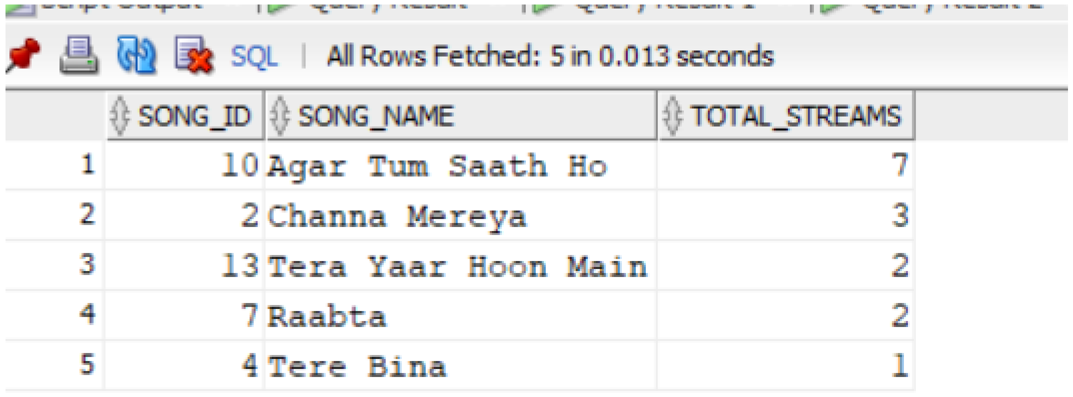
```
   INSERT INTO DURATION (subscription_plan_id, start_date, end_date)
   (SELECT
user_subscription_plan.subscription_plan_id,user_subscription_plan.start_date,a
dd_months(user_subscription_plan.start_date, subscription_plan.plan_duration)
AS end_date FROM (user_subscription_plan LEFT JOIN subscription_plan ON
user_subscription_plan.subscription_plan_id=subscription_plan.subscription_pla
n_id));

   SELECT * FROM DURATION;
   DESC DURATION;
```

TESTING QUERIES

⇨ TOP 5 TRENDING SONGS ON APP

WE'RE CONSIDERING A TRENDING SONG AS THE MOST POPULAR (STREAMED) TRACK AMONG THOSE UPLOADED IN THE LAST TWO MONTHS.

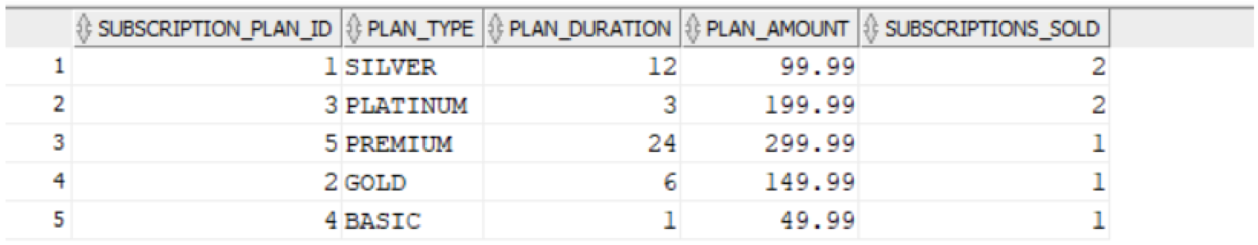
```
SELECT TABLE1.SONG_ID , SONG.SONG_NAME , TABLE1.STREAMS_IN_INDIA
FROM
( SELECT PLAYED_HISTORY.SONG_ID,COUNT(PLAYED_HISTORY.SONG_ID) AS
STREAMS_IN_INDIA
   FROM PLAYED_HISTORY,SONG
      WHERE PLAYED_HISTORY.SONG_ID=SONG.SONG_ID AND
SONG.DATE_OF_UPLOAD > ADD_MONTHS(SYSDATE,-2)
         GROUP BY PLAYED_HISTORY.SONG_ID ORDER BY STREAMS_IN_INDIA
DESC FETCH FIRST 5 ROWS ONLY
)TABLE1,
SONG WHERE SONG.SONG_ID=TABLE1.SONG_ID;
```



	SONG_ID	SONG_NAME	TOTAL_STREAMS
1	10	Agar Tum Saath Ho	7
2	2	Channa Mereya	3
3	13	Tera Yaar Hoon Main	2
4	7	Raabta	2
5	4	Tere Bina	1

⇨ Identify the most popular subscription plans.

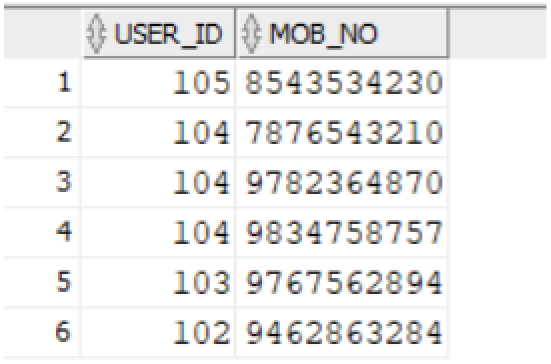
```
SELECT TABLE1.SUBSCRIPTION_PLAN_ID,PLAN_TYPE ,PLAN_DURATION
,PLAN_AMOUNT,SUBSCRIPTIONS_SOLD
FROM SUBSCRIPTION_PLAN,
(
        SELECT SUBSCRIPTION_PLAN_ID,
        COUNT(SUBSCRIPTION_PLAN_ID) AS SUBSCRIPTIONS_SOLD
        FROM USER_SUBSCRIPTION_PLAN GROUP BY SUBSCRIPTION_PLAN_ID
        ORDER BY SUBSCRIPTIONS_SOLD DESC
) TABLE1
WHERE SUBSCRIPTION_PLAN.SUBSCRIPTION_PLAN_ID =
TABLE1.SUBSCRIPTION_PLAN_ID;
```



	SUBSCRIPTION_PLAN_ID	PLAN_TYPE	PLAN_DURATION	PLAN_AMOUNT	SUBSCRIPTIONS_SOLD
1	1	SILVER	12	99.99	2
2	3	PLATINUM	3	199.99	2
3	5	PREMIUM	24	299.99	1
4	2	GOLD	6	149.99	1
5	4	BASIC	1	49.99	1

⇨ DISPLAY CONTACT DETAILS OF USERS INACTIVE FOR THE PAST 3 MONTHS TO REQUEST FEEDBACK

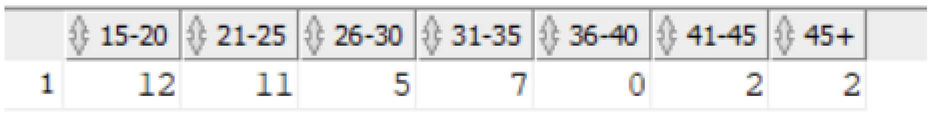
```
 SELECT USER_ID , MOB_NO
FROM CONTACT_DETAILS
WHERE USER_ID NOT IN
(
         SELECT USER_ID FROM PLAYED_HISTORY
         WHERE DATE_AND_TIME >= ADD_MONTHS (SYSDATE,-3)
);
```



	USER_ID	MOB_NO
1	105	8543534230
2	104	7876543210
3	104	9782364870
4	104	9834758757
5	103	9767562894
6	102	9462863284

⇨ Show the distribution of app users by age group.

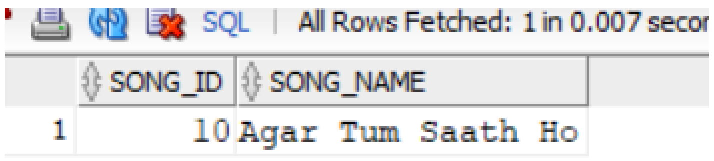
```
SELECT
   SUM(CASE WHEN USER_AGE BETWEEN 15 AND 20 THEN 1 ELSE 0 END) AS
"15-20",
   SUM(CASE WHEN USER_AGE BETWEEN 21 AND 25 THEN 1 ELSE 0 END) AS
"21-25",
   SUM(CASE WHEN USER_AGE BETWEEN 26 AND 30 THEN 1 ELSE 0 END) AS
"26-30",
   SUM(CASE WHEN USER_AGE BETWEEN 31 AND 35 THEN 1 ELSE 0 END) AS
"31-35",
   SUM(CASE WHEN USER_AGE BETWEEN 36 AND 40 THEN 1 ELSE 0 END) AS
"36-40",
   SUM(CASE WHEN USER_AGE BETWEEN 41 AND 45 THEN 1 ELSE 0 END) AS
"41-45",
   SUM(CASE WHEN USER_AGE > 45 THEN 1 ELSE 0 END) AS "45+"
FROM
   USER_DETAILS;
```



	15-20	21-25	26-30	31-35	36-40	41-45	45+
1	12	11	5	7	0	2	2

⇨ Show the most popular song by Arijit Singh.

```
SELECT SONG.SONG_ID,SONG_NAME FROM SONG,
(
        SELECT PLAYED_HISTORY.SONG_ID FROM
        PLAYED_HISTORY LEFT JOIN SONG
        ON SONG.SONG_ID=PLAYED_HISTORY.SONG_ID
        WHERE SINGER_ID=(SELECT ARTIST_ID FROM ARTIST WHERE
        ARTIST_NAME='ARIJIT SINGH')
        GROUP BY PLAYED_HISTORY.SONG_ID
        ORDER BY COUNT(PLAYED_HISTORY.SONG_ID) DESC
        FETCH FIRST 1 ROWS ONLY
) TABLE1
WHERE SONG.SONG_ID=TABLE1.SONG_ID;
```

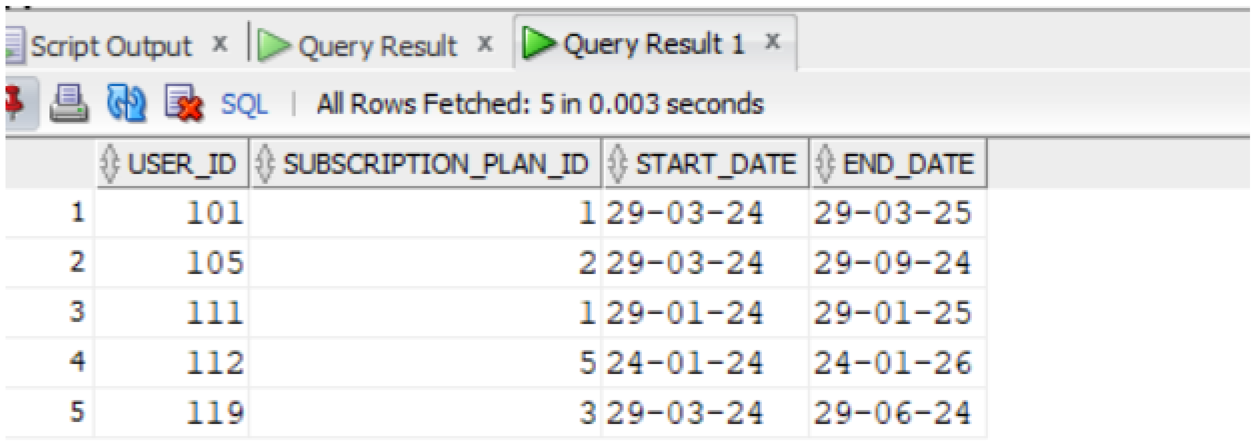


SQL | All Rows Fetched: 1 in 0.007 secor

	SONG_ID	SONG_NAME
1	10	Agar Tum Saath Ho

⇨ Show the active user subscriptions.

```
SELECT USER_SUBSCRIPTION_PLAN.USER_ID ,
USER_SUBSCRIPTION_PLAN.SUBSCRIPTION_PLAN_ID,
USER_SUBSCRIPTION_PLAN.START_DATE,
DURATION.END_DATE
FROM USER_SUBSCRIPTION_PLAN , DURATION
WHERE USER_SUBSCRIPTION_PLAN.SUBSCRIPTION_PLAN_ID =
DURATION.SUBSCRIPTION_PLAN_ID AND
USER_SUBSCRIPTION_PLAN.START_DATE=DURATION.START_DATE AND
END_DATE >= SYSDATE;
```



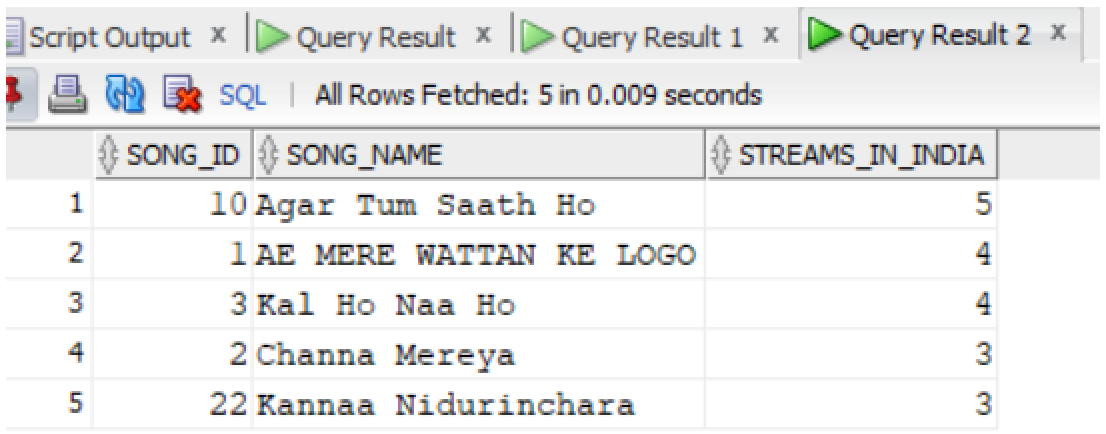
Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 5 in 0.003 seconds

	USER_ID	SUBSCRIPTION_PLAN_ID	START_DATE	END_DATE
1	101	1	29-03-24	29-03-25
2	105	2	29-03-24	29-09-24
3	111	1	29-01-24	29-01-25
4	112	5	24-01-24	24-01-26
5	119	3	29-03-24	29-06-24

⇨ TOP 5 POPULAR SONGS IN 'INDIA'

```
   SELECT TABLE1.SONG_ID , SONG.SONG_NAME , TABLE1.STREAMS_IN_INDIA
FROM
   ( SELECT SONG_ID,COUNT(SONG_ID) AS STREAMS_IN_INDIA
      FROM PLAYED_HISTORY,USER_DETAILS
      WHERE PLAYED_HISTORY.USER_ID = USER_DETAILS.USER_ID
       AND USER_DETAILS.NATION='INDIA'
      GROUP BY SONG_ID
      ORDER BY STREAMS_IN_INDIA DESC FETCH FIRST 5 ROWS ONLY
   )TABLE1,
   SONG WHERE SONG.SONG_ID=TABLE1.SONG_ID;
```



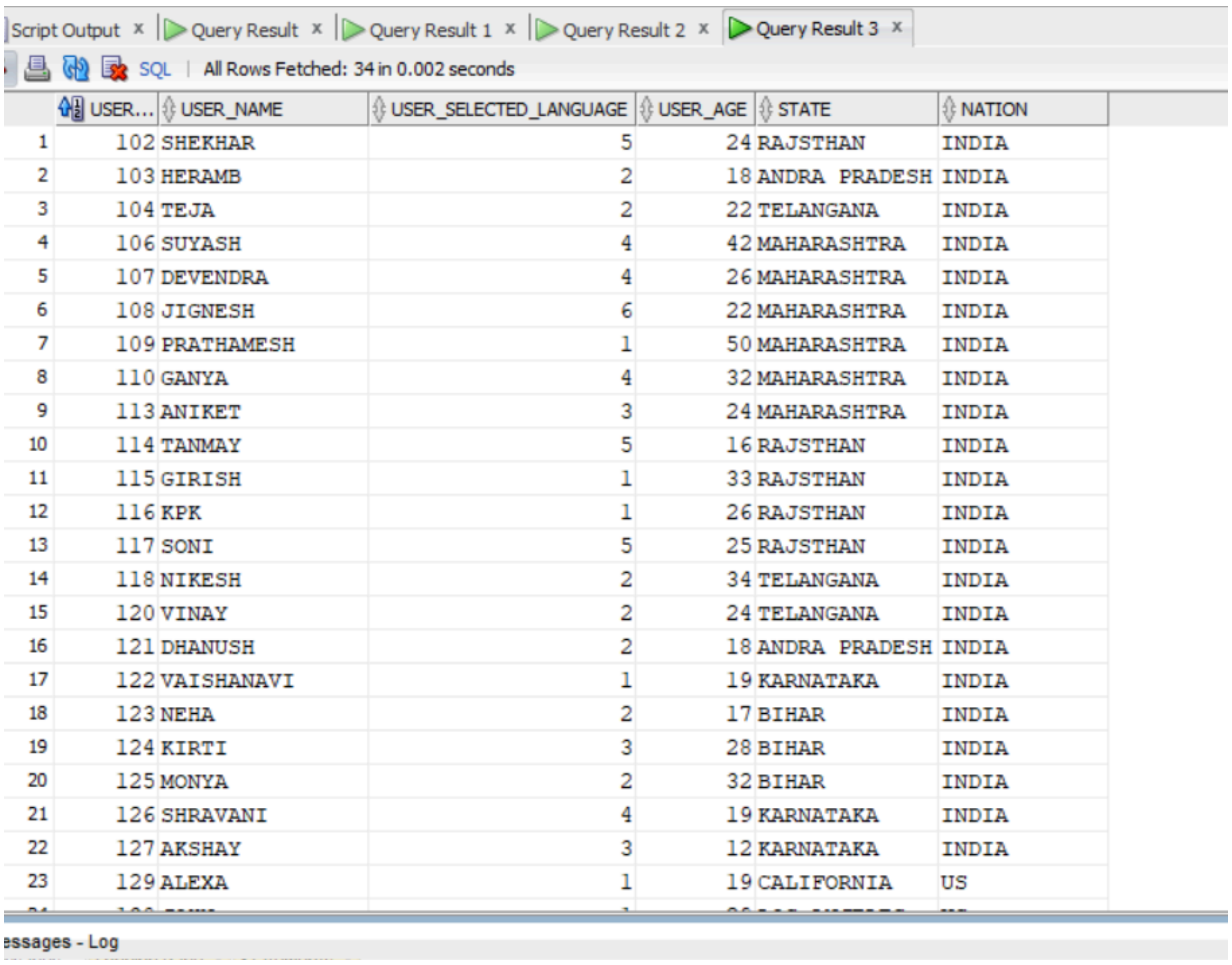
Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 5 in 0.009 seconds

	SONG_ID	SONG_NAME	STREAMS_IN_INDIA
1	10	Agar Tum Saath Ho	5
2	1	AE MERE WATTAN KE LOGO	4
3	3	Kal Ho Naa Ho	4
4	2	Channa Mereya	3
5	22	Kannaa Nidurinchara	3

⇨ Users who haven't subscribed

```
SELECT * FROM USER_DETAILS
WHERE USER_ID
    NOT IN
        (SELECT USER_ID FROM USER_SUBSCRIPTION_PLAN);
```



All Rows Fetched: 34 in 0.002 seconds

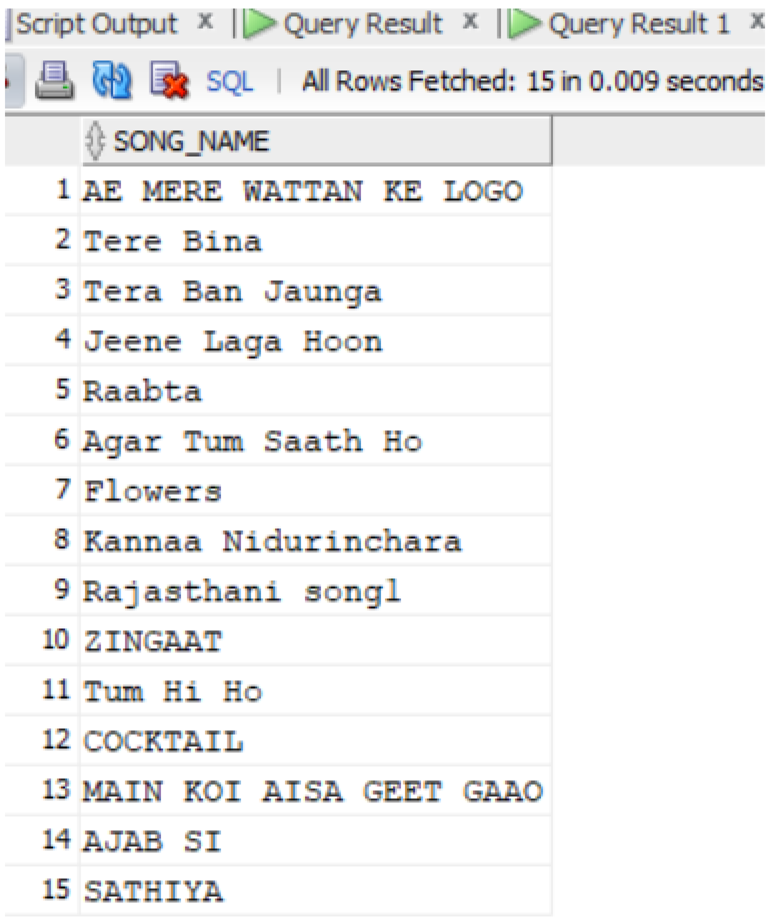
	USER...	USER_NAME	USER_SELECTED_LANGUAGE	USER_AGE	STATE	NATION
1	102	SHEKHAR	5	24	RAJSTHAN	INDIA
2	103	HERAMB	2	18	ANDRA PRADESH	INDIA
3	104	TEJA	2	22	TELANGANA	INDIA
4	106	SUYASH	4	42	MAHARASHTRA	INDIA
5	107	DEVENDRA	4	26	MAHARASHTRA	INDIA
6	108	JIGNESH	6	22	MAHARASHTRA	INDIA
7	109	PRATHAMESH	1	50	MAHARASHTRA	INDIA
8	110	GANYA	4	32	MAHARASHTRA	INDIA
9	113	ANIKET	3	24	MAHARASHTRA	INDIA
10	114	TANMAY	5	16	RAJSTHAN	INDIA
11	115	GIRISH	1	33	RAJSTHAN	INDIA
12	116	KPK	1	26	RAJSTHAN	INDIA
13	117	SONI	5	25	RAJSTHAN	INDIA
14	118	NIKESH	2	34	TELANGANA	INDIA
15	120	VINAY	2	24	TELANGANA	INDIA
16	121	DHANUSH	2	18	ANDRA PRADESH	INDIA
17	122	VAISHANAVI	1	19	KARNATAKA	INDIA
18	123	NEHA	2	17	BIHAR	INDIA
19	124	KIRTI	3	28	BIHAR	INDIA
20	125	MONYA	2	32	BIHAR	INDIA
21	126	SHRAVANI	4	19	KARNATAKA	INDIA
22	127	AKSHAY	3	12	KARNATAKA	INDIA
23	129	ALEXA	1	19	CALIFORNIA	US

essages - Log

⇨ CREATE A PERSONALIZED PLAYLIST FOR USER 115 ACCORDING GENRE HE LISTENS MOST;

FLOW --> FIND THE MOST PLAYED GENRE BY THE USER AND SUGGEST MORE SONG IN THE SAME GENRE;

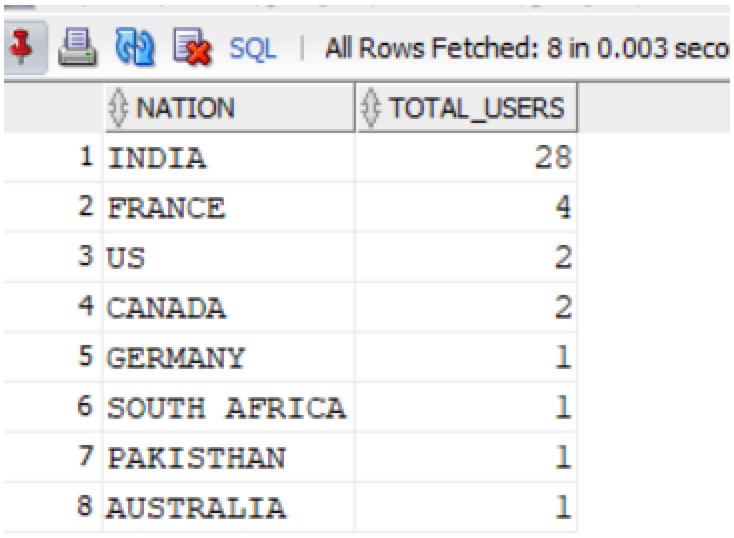
```
SELECT SONG_NAME FROM SONG WHERE GENRE =
(
SELECT GENRE FROM
PLAYED_HISTORY LEFT JOIN SONG ON
PLAYED_HISTORY.SONG_ID=SONG.SONG_ID
WHERE USER_ID=115
GROUP BY GENRE
FETCH FIRST 1 ROWS ONLY
);
```



```
All Rows Fetched: 15 in 0.009 seconds

   SONG_NAME
1  AE MERE WATTAN KE LOGO
2  Tere Bina
3  Tera Ban Jaunga
4  Jeene Laga Hoon
5  Raabta
6  Agar Tum Saath Ho
7  Flowers
8  Kannaa Nidurinchara
9  Rajasthani song1
10 ZINGAAT
11 Tum Hi Ho
12 COCKTAIL
13 MAIN KOI AISA GEET GAAO
14 AJAB SI
15 SATHIYA
```

⇨ Which country contributes the highest number of users to our platform?

```
SELECT NATION,COUNT(NATION) AS TOTAL_USERS
   FROM USER_DETAILS GROUP BY NATION
      ORDER BY COUNT(NATION) DESC;
```



SQL | All Rows Fetched: 8 in 0.003 seco

	NATION	TOTAL_USERS
1	INDIA	28
2	FRANCE	4
3	US	2
4	CANADA	2
5	GERMANY	1
6	SOUTH AFRICA	1
7	PAKISTHAN	1
8	AUSTRALIA	1

CONCLUSION

Analyzing MUSIC STREAMING APP database is super valuable. It helps understand what music people love, which genres are hot, and where trends are headed. With millions of users, it's like a goldmine of information.

First off, you can see what people are listening to most. This helps us to suggest new music and create awesome playlists that match your taste. It's like having a personal DJ!

Plus, it's great for spotting trends. If a certain genre or artist starts blowing up, our data shows it. This is huge for music labels and artists trying to make it big. And it's not just about global trends. You can see what's popular in different regions. That way, artists can plan tours and marketers can target specific areas.

This helps with things like planning tours and connecting with fans.

For advertisers, it's a goldmine. They can create super-targeted ads based on user data. This means ads that actually resonate with people, leading to better results.

On top of that, researchers love it. They can dive deep into the data to understand broader trends in music consumption and culture. It's like a window into how society vibes with music.

Even musicologists and scholars dig it. They use the data to study music's cultural and historical impact. It's like a time machine, showing how music trends have evolved over time.

In a nutshell, MUSIC STREAMING APP database is a game-changer. It's not just about streaming music—it's about understanding people's connection to it. And that's pretty cool.

OUTCOMES

Certainly, here are five outcomes of accessing our database:

Personalized Recommendations: Utilizing user data and listening history to generate personalized music recommendations, playlists, and suggestions tailored to individual preferences.

Analytics and Insights: Analyzing trends in music consumption, popular genres, emerging artists, and regional variations to gain insights into consumer behavior and industry trends.

Content Creation: Creating playlists, albums, and featured content based on user engagement, popularity, and editorial curation strategies to enhance user experience and promote discovery.

Targeted Advertising: Leveraging demographic, geographic, and behavioral data to deliver targeted advertising campaigns to users based on their interests, preferences, and listening habits.

Platform Development: Informing the development of new features, services, and products by understanding user interactions, feedback, and usage patterns within the app ecosystem to improve overall platform performance and user satisfaction.