Department of Computer Science & Information Technology

CT-261 Database Management Systems

Complex Computing Problem (CCP)

Choose a problem in any application domain of computing and ***develop*** its solution through an effective use of database management system principles. The solution must be in the form of a well-documented and well-formatted report. It must be explained by employing in-depth computing or domain knowledge, and an approach that is based on well-founded principles.

The following instructions must be taken into consideration while preparing the project report. The report must reflect conceptual thinking and must properly be modularized according to functionality. The report of the solution must be organized clearly and should:

- Describe the steps of your solution,
- The DBMS techniques used,
- The limitations of the adopted approach.

Complex Computing Problem Assessment Rubrics

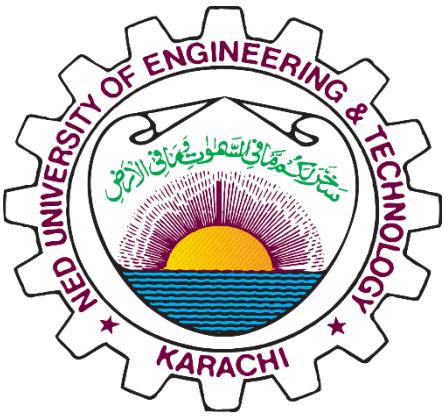
Criteria and Scales		
Good (10-9)	**Average (6-5)**	**Poor (1-0)**
Criterion 1 Problem Understanding: To what extent has the student has understood the problem? **(CP3: *Depth of Knowledge Required*)**		
The student has clearly understood the Problem.	The student has some clarity of the problem	The student has misunderstood the Problem.
Criterion 2 Requirement Identification: To what extent has the student outlined the database management systems principles required for solving the problem. ***(CP10: Requirement Identification*)**		
The student has clearly outlined the required database management systems principles.	The student has outlined some of the required database management systems principles.	The students have not outlined the database management systems principles.
Criterion 3 Clarity of Solution: To what extent has the student defined the solution of the problem. **(CP3: *Depth of Knowledge Required*)**		
The student has clearly defined the solution which works for all possible test cases.	The student has defined a solution which works well for a few test cases.	The student has not defined a correct solution.

Total marks: _____/10_____

Teacher's signature: _____

NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Computer Science and Information Technology



HERTZ: A MUSIC STREAMING PLATFORM

DBMS LAB PROJECT REPORT

Submitted by:

Muqaddas Ali CT-21001

Fatima Nadeem CT-21011

Aamna Khalid CT-21012

Aisha Shahzad CT-21015

Submitted to:

Sir Rohail Qamar

ORGANIZATION INTRODUCTION: HERTZ

Hertz, a music streaming service, is an online platform that allows users to listen to music over the internet without having to download the files. Users can access a vast library of songs, albums and playlists through dedicated apps or web browsers on various devices. These services offer personalized recommendations and both free/paid subscription options. It is convenient, with on-demand access, and a wide range of content.

PROBLEM OF ORGANIZATION

Music streaming services are becoming increasingly popular as a means of acquiring and consuming music. A Database Management System (DBMS) can help address the following mentioned issues that a music streaming platform may face:

1. Slow Performance And Latency Issues: A well-designed DBMS optimizes query execution, indexes data for faster retrieval, and utilizes efficient caching mechanisms. By optimizing data access and storage, a DBMS can significantly improve performance and reduce latency. A well-designed DBMS can ensure efficient and effective data management, allowing for high performance and responsiveness.

2. Lack Of Personalized Recommendations: A DBMS can leverage advanced algorithms and data processing capabilities to analyze user data and generate personalized recommendations. By efficiently storing and querying user preferences and behavior data, a DBMS can enable the generation of tailored recommendations for each user.The DBMS enables features like personalization and recommendations, allowing us to analyse user data and provide tailored content recommendations.

3. Difficulty In Tracking And Managing User Accounts: A DBMS provides robust user management capabilities, allowing for secure authentication, access control, and user account tracking. It enables efficient user account creation, modification, and deletion, simplifying user management processes.It can seamlessly store and manage user profiles, playback history, playlists, and other personalized content, enabling a seamless user experience.

4. Higher Risk Of Data Loss: A reliable DBMS ensures data integrity and durability through features like transaction management, data backup, and disaster recovery mechanisms. It implements ACID (Atomicity, Consistency, Isolation, Durability) properties to guarantee data reliability and minimize the risk of data loss.

5. Inability To Process Or Handle Large Amounts Of Data: A DBMS is designed to handle large-scale data processing and storage. It employs scalable architectures, distributed computing techniques, and parallel processing capabilities to efficiently manage and process vast amounts of data, enabling organizations to handle big data challenges.With millions of subscribers accessing these platforms daily, a robust and scalable database management system is essential for storing and managing the vast amounts of data generated by these users.

By implementing a robust and well-optimized DBMS, organizations can address these issues effectively, ensuring improved performance, personalized recommendations, streamlined user management, data integrity, and scalability to handle large datasets.

SCENARIO OF ORGANIZATION

ENTITIES:

1. **USERS**
   Represents the individuals using the music platform. It includes the attributes USER ID, NAME, USER_NAME, EMAIL and DATE_OF_BIRTH.

2. **GENRES**
   Represents different genres of music. It includes the attributes GENRE_NAME and GENRE_DESCRIPTION.

3. **ARTISTS**
   Represents the music artists or bands. It includes the attributes ARTIST_ID, ARTIST_NAME, ARTIST_BIO, MAIN_GENRE and LABEL_ID.

4. **ALBUMS**
   Represents the albums released by artists. It includes the attributes ALBUM_ID, ARTIST_ID, ALBUM_TITLE, and RELEASE_DATE.

5. **SONGS**
   Represents individual songs. It includes the attributes SONG_ID, SONG_TITLE, ARTIST_ID, ALBUM_ID, DURATION_IN_SECONDS, RELEASE_DATE and GENRE.

6. **PLAYLISTS**
   Represents curated collections of songs created by users. It includes the attributes PLAYLIST_ID, PLAYLIST_NAME, USER_ID, CREATION_DATE and PLAYLIST_DESCRIPTION.

7. **PLAYLISTS_SONGS**
   Represents the relationship between playlists and songs. It includes the attributes PLAYLIST_ID, SONG_ID, and SONG_POSITION.

8. **USER_LIBRARY**
   Represents the collection of albums and artists saved by users in their personal library. It includes the attributes LIBRARY_POSITION_ID, LIBRARY_ID, USER_ID, ALBUM_ID, ARTIST_ID and DATE_ADDED.

9. **PLAYBACK_HISTORY**
   Tracks the history of songs played by users. It includes the attributes PLAYBACK_ID, USER_ID, SONG_ID, PLAYLIST_ID, and LISTENED_DATE.

10. **RECOMMENDATIONS**
    Represents personalized song recommendations based on songs in the playlists. It includes the attributes RECOMMENDATION_ID, RECOMMENDED_SONG_ID, and PLAYLIST_ID.

11. **SUBSCRIPTION_PLANS**

    Represents different subscription plans offered to the users by the music platform. It includes the attributes PLAN_ID, PLAN_NAME, MONTHLY_COST_IN_RS, and PLAN_DESCRIPTION.

12. **SUBSCRIPTIONS**

    Represents the subscription details of users. It includes attributes: SUBSCRIPTION_ID, USER_ID, PLAN_ID, START_DATE, and EXPIRATION_DATE.

13. **PAYMENTS**

    Tracks the payment information for user subscriptions. It includes the attributes PAYMENT_ID, SUBSCRIPTION_ID, PAYMENT_DUE_DATE, PAYMENT_DATE, AMOUNT_PAID and PAYMENT_STATUS.

14. **USER_REVIEWS**

    Represents users' reviews and ratings for songs. It includes the attributes REVIEW_ID, USER_ID, SONG_ID, RATING, and COMMENTS.

15. **MUSIC_CHARTS**

    Represents music charts showcasing popular songs. It includes the attributes CHART_ID, CHART_NAME, CHART_TYPE, GENRE, CHART_DESCRIPTION and CREATION_DATE.

16. **CHART_SONGS**

    Represents the songs that are included in some music charts. It includes the attributes CHART_ID, SONG_ID and RANK, indicating which songs are included in each chart at what rank.

17. **MUSIC_LABELS**

    Represents record labels or music companies. It includes the attributes LABEL_ID, LABEL_NAME, FOUNDER and YEAR_FOUNDED.

18. **MUSIC_LICENSE**

    Represents the types of music licenses available. It includes the attributes LICENSE ID, LICENSE_NAME, FEE_IN_RS, FEE_PERIOD and LICENSE_AGREEMENT.

19. **ALBUM_LICENSE**

    Represents the licensing agreements that an album has. It includes the attributes ALBUM_ID, LICENSE_ID, START_DATE and END_DATE.

20. **FOLLOWERS**

    Represents the relationship between users and artists, indicating which artists are followed by which users. It includes the attributes ARTIST_ID, FOLLOWER_ID and DATE_FOLLOWED.

RELATIONSHIPS:

1. **Have**
   USERS have PLAYBACK_HISTORY
   SUBSCRIPTIONS have SUBSCRIPTION_PLANS
   USERS have SUBSCRIPTIONS
   ARTISTS have FOLLOWERS
   ARTISTS have GENRES
   SONGS have GENRES
   SONGS have USER_REVIEWS
   MUSIC_CHARTS have GENRES
   USER have USER_LIBRARY

2. **Contain**
   MUSIC_CHARTS contain CHART_SONGS
   RECOMMENDATIONS contain SONGS
   ALBUMS contain SONGS
   USER_LIBRARY contain ALBUMS

3. **Are in**
   USERS are FOLLOWERS
   SONGS are in PLAYLIST_SONGS
   SONGS are in CHART_SONGS
   ALBUM_LICENSE are MUSIC_LICENSE

4. **Get**
   PLAYLISTS get RECOMMENDATIONS

5. **Require**
   SUBSCRIPTIONS require PAYMENTS

6. **Release**
   ARTISTS release SONGS
   ARTISTS release ALBUMS

7. **Make**
   USERS make PLAYLISTS

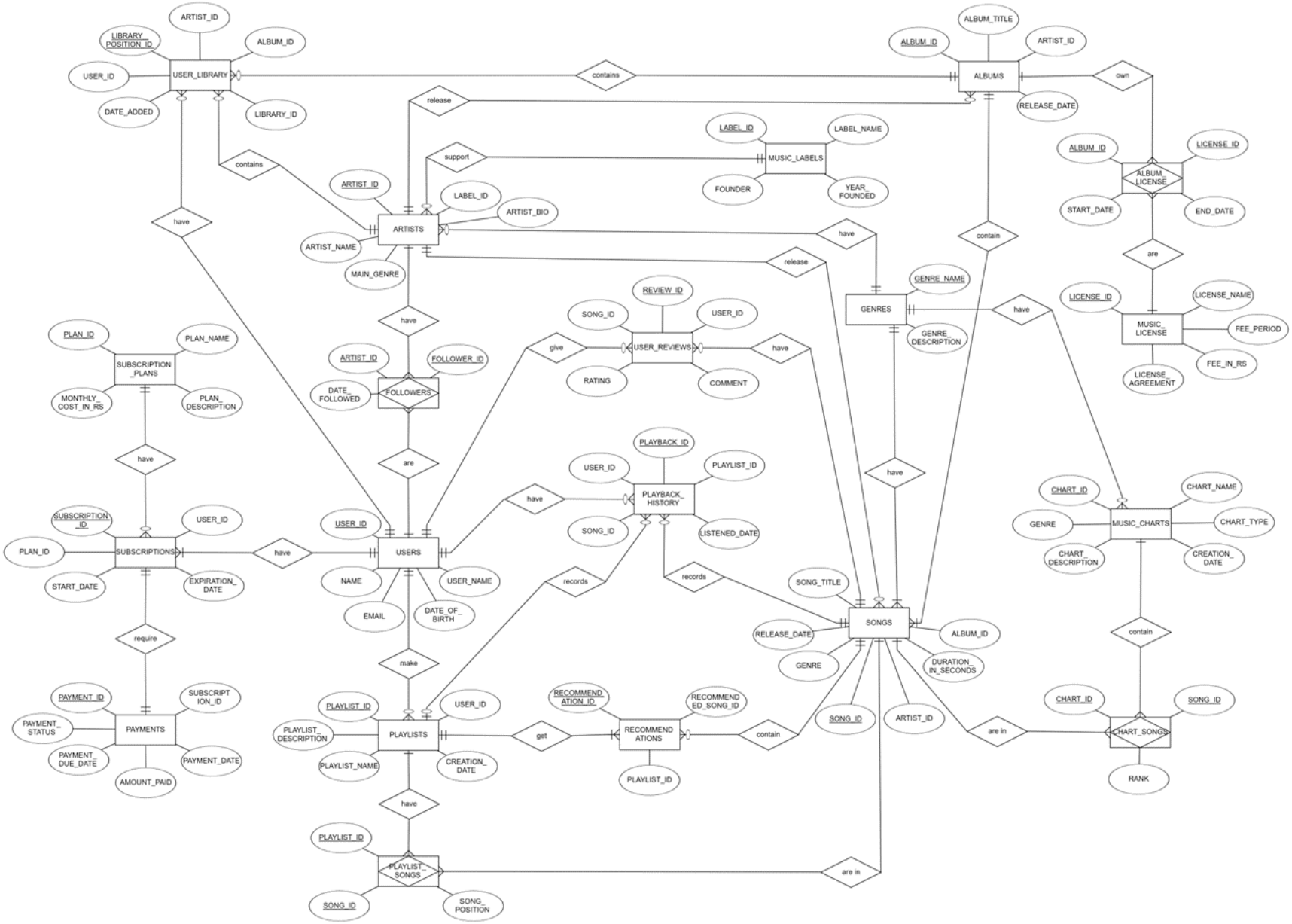
8. **Records**
   PLAYBACK_HISTORY records SONGS
   PLAYBACK_HISTORY records PLAYLISTS

9. **Own**
   ALBUMS own ALBUM_LICENSE

10. **Support**
    MUSIC_LABELS support ARTISTS

11. **Give**
    USERS give USER_REVIEWS

ENTITY-RELATIONSHIP DIAGRAM



DBMS TECHNIQUES USED

- Normalization
- Concurrency control
- Data validation and constraints

FOREIGN KEYS IN THE SCHEMA

ENTITY	FOREIGN KEY	REFERENCE TABLE
USERS	-	-
GENRES	-	-
MUSIC_LABELS	-	-
ARTISTS	MAIN_GENRE LABEL_ID	GENRES MUSIC_LABELS
ALBUMS	ARTIST_ID	ARTISTS
SONGS	ARTIST_ID ALBUM_ID GENRE	ARTISTS ALBUMS GENRES
PLAYLISTS	USER_ID	USERS
PLAYLIST_SONGS	PLAYLIST_ID SONG_ID	PLAYLISTS SONGS
USER_LIBRARY	USER_ID ARTIST_ID ALBUM_ID	USERS ARTISTS ALBUMS
PLAYBACK_HISTORY	USER_ID SONG_ID PLAYLIST_ID	USERS SONGS PLAYLISTS
RECOMMENDATIONS	RECOMMENDE_SONG_ID PLAYLIST_ID	SONGS PLAYLISTS
SUBSCRIPTION_PLANS	-	-
SUBSCRIPTIONS	USER_ID SUBSCRIPTION_ID	USERS SUBSCRIPTION_PLANS
PAYMENTS	SUBSCRIPTION_ID	SUBSCRIPTIONS
USER_REVIEWS	USERS_ID SONG_ID	USERS SONGS
MUSIC_CHARTS	GENRE	GENRES

CHART_SONGS	CHART_ID SONG_ID	MUSIC_CHARTS SONGS
MUSIC_LICENSE	-	-
ALBUM_LICENSE	ALBUM_ID LICENSE_ID	ALBUMS MUSIC_LICENSE
FOLLOWERS	ARTIST_ID FOLLOWER_ID	ARTISTS USERS

ENTITY AND REFERENTIAL INTEGRITY

The **Entity Integrity Rule** ensures that each entity in the database has a unique identifier, like a primary key, to identify each record individually. The **Referential Integrity Rule** ensures that a foreign key in one table points to a valid primary key in another table, helping to maintain data consistency by preventing accidental deletion or modification of related records.

To understand how these rules apply to our schema, we shall look at USERS and PLAYLISTS entities. The USERS table has an attribute called USER_ID and the PLAYLISTS table also has a USER_ID attribute. In this relation, the Entity Integrity Rule applies to USERS, while the Referential Integrity Rule applies to PLAYLISTS.

The USER_ID in the USERS table serves as a **primary key** for each user in the table. This means that each user in the table has a unique USER_ID.

The USER_ID in the PLAYLISTS table is a **foreign key** that references the primary key in the USERS table (USER_ID). This means that for each playlist, there must be a corresponding user record in the USERS table with the same user_ID value. This allows the PLAYLISTS table to link each playlist to a specific user, based on their shared user_ID value. Furthermore, the PLAYLISTS table also has a primary key of its own called PLAYLISTS_ID.

In this way Entity Integrity Rule is applied to each and every entity and Referential Integrity Rules is applied to every entity that has a foreign key.

GENERAL CONSTRAINTS

The general constraints used in this schema are primary keys, foreign keys, not null, unique, default values and check constraints.

1. **Primary Key Constraint**
   Primary key constraint is used in every table to ensure that every record has a unique identifier. For example, the “ALBUMS” table has a primary key “ALBUM_ID” that uniquely identifies each individual album record.

2. **Foreign Key Constraint**
   Foreign key constraint is used in every table in which an attribute from another table is used. This helps ensure that values in the dependent table refers to an existing value in the referenced table. For example, the “ALBUMS” table has a foreign key

"ARTIST_ID" that refers to the "ARTISTS" table, indicating that the album is associated with a specific artist.

3. **Not Null**
   Not Null constraint is used to ensure that a certain attribute must always have a value and can never be null. For example, in the "ALBUMS" table, the attribute "ALBUM_TITLE" has a Not Null constraint because every album always has a title.

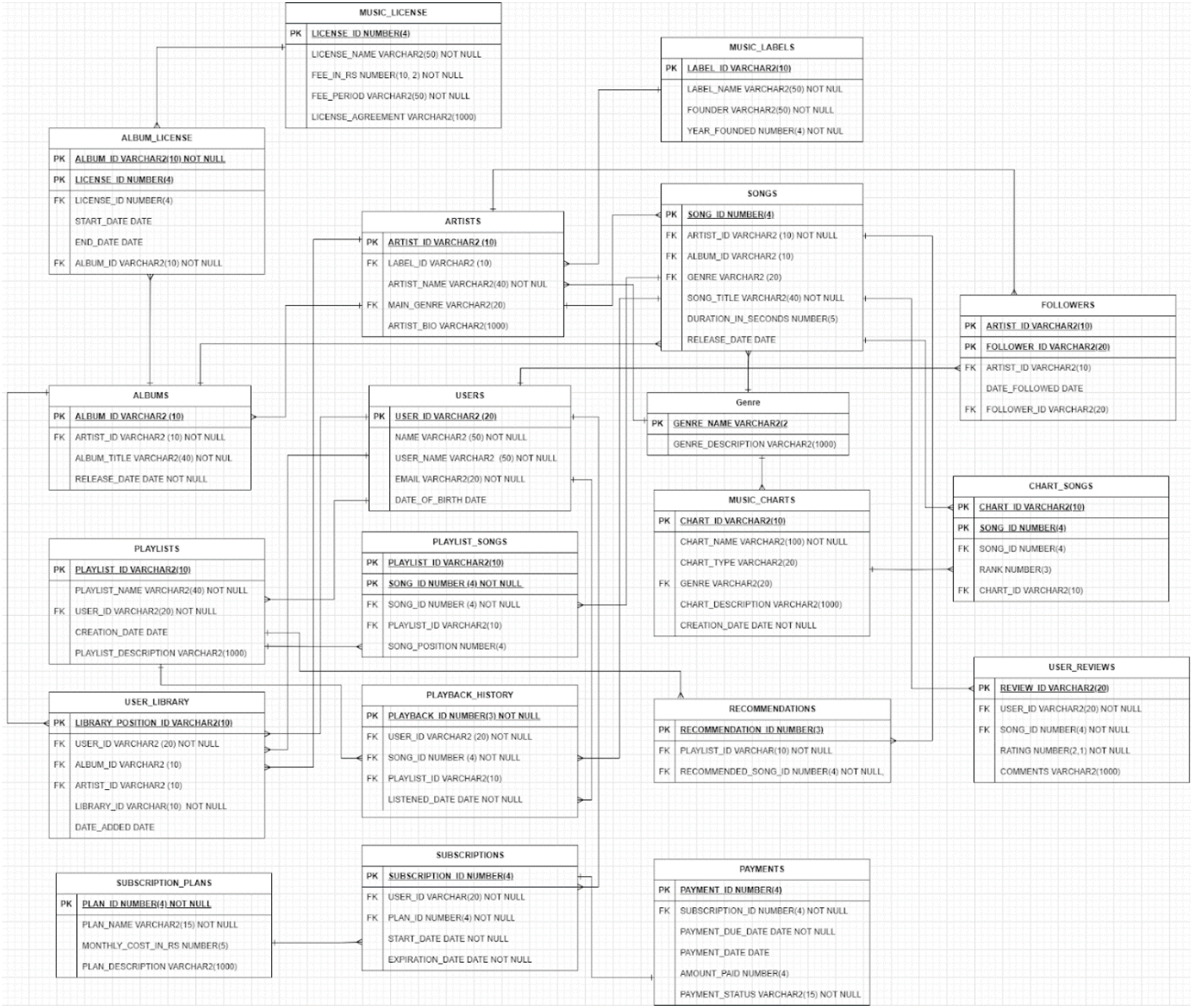
4. **Unique Constraint**
   Unique constraint is used to ensure that no two records in a table can have the same value for a certain attribute. It helps prevent duplicate data from entering the system. For example, the "USERS" table has a unique constraint for the "EMAIL" attribute as it is not possible for two users to have the same email address.

5. **Check Constraint**
   Check constraint is used to ensure that data is properly recorded. For example, in the "ALBUM_LICENSE" table date check is used to ensure that the license "END_DATE" is always greater than the license "START_DATE".

6. **Default Constraint**
   Default constraint is used to provide a default value for an attribute if one is not provided. For example, in the "MUSIC_CHART" table if the "CREATION_DATE" has not been inputted then the system date will automatically be assigned to it.

PHYSICAL ENTITY RELATIONSHIP DIAGRAM

All tables are in 3rd normal form.


MUSIC_LICENSE
PK LICENSE_ID NUMBER(4)
LICENSE_NAME VARCHAR2(50) NOT NULL
FEE_IN_RS NUMBER(10, 2) NOT NULL
FEE_PERIOD VARCHAR2(50) NOT NULL
LICENSE_AGREEMENT VARCHAR2(1000)
MUSIC_LABELS
PK LABEL_ID VARCHAR2(10)
LABEL_NAME VARCHAR2(50) NOT NUL
FOUNDER VARCHAR2(50) NOT NULL
YEAR_FOUNDED NUMBER(4) NOT NUL
ALBUM_LICENSE
PK ALBUM_ID VARCHAR2(10) NOT NULL
PK LICENSE_ID NUMBER(4)
FK LICENSE_ID NUMBER(4)
START_DATE DATE
END_DATE DATE
FK ALBUM_ID VARCHAR2(10) NOT NULL
ARTISTS
PK ARTIST_ID VARCHAR2 (10)
FK LABEL_ID VARCHAR2 (10)
ARTIST_NAME VARCHAR2(40) NOT NUL
FK MAIN_GENRE VARCHAR2(20)
ARTIST_BIO VARCHAR2(1000)
SONGS
PK SONG_ID NUMBER(4)
FK ARTIST_ID VARCHAR2 (10) NOT NULL
FK ALBUM_ID VARCHAR2 (10)
FK GENRE VARCHAR2 (20)
SONG_TITLE VARCHAR2(40) NOT NULL
DURATION_IN_SECONDS NUMBER(5)
RELEASE_DATE DATE
FOLLOWERS
PK ARTIST_ID VARCHAR2(10)
PK FOLLOWER_ID VARCHAR2(20)
FK ARTIST_ID VARCHAR2(10)
DATE_FOLLOWED DATE
FK FOLLOWER_ID VARCHAR2(20)
ALBUMS
PK ALBUM_ID VARCHAR2 (10)
FK ARTIST_ID VARCHAR2 (10) NOT NULL
ALBUM_TITLE VARCHAR2(40) NOT NUL
RELEASE_DATE DATE NOT NULL
USERS
PK USER_ID VARCHAR2 (20)
NAME VARCHAR2 (50) NOT NULL
USER_NAME VARCHAR2 (50) NOT NULL
EMAIL VARCHAR2(20) NOT NULL
DATE_OF_BIRTH DATE
Genre
PK GENRE_NAME VARCHAR2(2
GENRE_DESCRIPTION VARCHAR2(1000)
CHART_SONGS
PK CHART_ID VARCHAR2(10)
PK SONG_ID NUMBER(4)
FK SONG_ID NUMBER(4)
RANK NUMBER(3)
FK CHART_ID VARCHAR2(10)
MUSIC_CHARTS
PK CHART_ID VARCHAR2(10)
CHART_NAME VARCHAR2(100) NOT NULL
CHART_TYPE VARCHAR2(20)
FK GENRE VARCHAR2(20)
CHART_DESCRIPTION VARCHAR2(1000)
CREATION_DATE DATE NOT NULL
PLAYLISTS
PK PLAYLIST_ID VARCHAR2(10)
PLAYLIST_NAME VARCHAR2(40) NOT NULL
FK USER_ID VARCHAR2(20) NOT NULL
CREATION_DATE DATE
PLAYLIST_DESCRIPTION VARCHAR2(1000)
PLAYLIST_SONGS
PK PLAYLIST_ID VARCHAR2(10)
PK SONG_ID NUMBER (4) NOT NULL
FK SONG_ID NUMBER (4) NOT NULL
FK PLAYLIST_ID VARCHAR2(10)
SONG_POSITION NUMBER(4)
USER_REVIEWS
PK REVIEW_ID VARCHAR2(20)
FK USER_ID VARCHAR2(20) NOT NULL
FK SONG_ID NUMBER(4) NOT NULL
RATING NUMBER(2,1) NOT NULL
COMMENTS VARCHAR2(1000)
USER_LIBRARY
PK LIBRARY_POSITION_ID VARCHAR2(10)
FK USER_ID VARCHAR2 (20) NOT NULL
FK ALBUM_ID VARCHAR2 (10)
FK ARTIST_ID VARCHAR2 (10)
LIBRARY_ID VARCHAR(10) NOT NULL
DATE_ADDED DATE
PLAYBACK_HISTORY
PK PLAYBACK_ID NUMBER(3) NOT NULL
FK USER_ID VARCHAR2 (20) NOT NULL
FK SONG_ID NUMBER (4) NOT NULL
FK PLAYLIST_ID VARCHAR2(10)
LISTENED_DATE DATE NOT NULL
RECOMMENDATIONS
PK RECOMMENDATION_ID NUMBER(3)
FK PLAYLIST_ID VARCHAR(10) NOT NULL
FK RECOMMENDED_SONG_ID NUMBER(4) NOT NULL
SUBSCRIPTIONS
PK SUBSCRIPTION_ID NUMBER(4)
FK USER_ID VARCHAR(20) NOT NULL
FK PLAN_ID NUMBER(4) NOT NULL
START_DATE DATE NOT NULL
EXPIRATION_DATE DATE NOT NULL
PAYMENTS
PK PAYMENT_ID NUMBER(4)
FK SUBSCRIPTION_ID NUMBER(4) NOT NULL
PAYMENT_DUE_DATE DATE NOT NULL
PAYMENT_DATE DATE
AMOUNT_PAID NUMBER(4)
PAYMENT_STATUS VARCHAR2(15) NOT NULL
SUBSCRIPTION_PLANS
PK PLAN_ID NUMBER(4) NOT NULL
PLAN_NAME VARCHAR2(15) NOT NULL
MONTHLY_COST_IN_RS NUMBER(5)
PLAN_DESCRIPTION VARCHAR2(1000)

TYPES OF USERS AND THEIR PRIVILEGES

USERS			
Fatima Nadeem	**Aisha Shahzad**	**Aamna Khalid**	**Muqaddas Ali**
Database Administrator	Application Developer	Application Developer	Application Developer
Manages and maintains databases for efficient and secure data operations.	Designs, develops, and maintains software applications for different platforms.	Designs, develops, and maintains software applications for different platforms.	Designs, develops, and maintains software applications for different platforms.

LIMITATIONS

- The USER_LIBRARY table only refers to playlists, albums and artists, not songs. This is because while creating the ERD on ERDPlus, an entity can only act as the referenced table for 8 other tables and the SONG_ID attribute from the SONGS table is already being referenced in 8 different tables.

- The schema does not include mechanisms for auditing or tracking changes made to the data. This can make it challenging to maintain an audit trail and track historical changes, which may be important for compliance or data governance purposes.

- The schema does not explicitly define indexes on the tables, which can impact query performance, especially for frequently accessed columns or complex join operations.

QUERIES PERFORMED *(schema given below)*

JOINING

To show which song is in which chart on which rank:

```
SELECT  C.CHART_ID, M.CHART_NAME, S.SONG_ID, S.SONG_TITLE, C.RANK, M.CHART_DESCRIPTION
FROM MUSIC_CHARTS M
INNER JOIN CHART_SONGS C
ON C.CHART_ID=M.CHART_ID
INNER JOIN SONGS S
ON C.SONG_ID = S.SONG_ID
ORDER BY C.CHART_ID, C.RANK;
```

Retrieves the name and email of users who follow a specific artist:

```
SELECT U.NAME, U.EMAIL
FROM USERS U
JOIN FOLLOWERS F
ON U.USER_ID = F.FOLLOWER_ID
WHERE F.ARTIST_ID = 'HS-1994';
```

Retrieves the names of users who have reviewed songs by a specific artist

```
SELECT U.NAME
FROM USERS U
JOIN USER_REVIEWS UR ON U.USER_ID = UR.USER_ID
JOIN SONGS S ON UR.SONG_ID = S.SONG_ID
WHERE S.ARTIST_ID = 'LDR-1985';
```

SUBQUERY

Retrieve the names of users who have a higher subscription cost than the average monthly cost:

```
SELECT NAME FROM USERS
WHERE USER_ID IN (
        SELECT USER_ID FROM SUBSCRIPTIONS
        JOIN SUBSCRIPTION_PLANS
        ON SUBSCRIPTIONS.PLAN_ID = SUBSCRIPTION_PLANS.PLAN_ID
        WHERE MONTHLY_COST_IN_RS > (
                SELECT AVG(MONTHLY_COST_IN_RS) FROM SUBSCRIPTION_PLANS
        )
);
```

Retrieve the names of albums that have songs in a specific genre:

```
SELECT ALBUM_TITLE
FROM ALBUMS
WHERE ALBUM_ID IN (
            SELECT ALBUM_ID
            FROM SONGS
            WHERE GENRE = 'Pop'
);
```

Retrieve the names of users who have subscribed to a specific plan:

```
SELECT NAME
FROM USERS
WHERE USER_ID IN (
  SELECT USER_ID
  FROM SUBSCRIPTIONS
  WHERE PLAN_ID IN (
    SELECT PLAN_ID
    FROM SUBSCRIPTION_PLANS
    WHERE PLAN_NAME = 'Premium'
  )
);
```

GROUP FUNCTION

Retrieves the playlist names along with the number of songs in each playlist:

```
SELECT P.PLAYLIST_NAME, COUNT(PS.SONG_ID) AS SONG_COUNT
FROM PLAYLISTS P
LEFT JOIN PLAYLIST_SONGS PS ON P.PLAYLIST_ID = PS.PLAYLIST_ID
GROUP BY P.PLAYLIST_NAME;
```

Calculates the total duration of songs in each playlist:

```
SELECT PS.PLAYLIST_ID, SUM(S.DURATION_IN_SECONDS) AS TOTAL_DURATION
FROM PLAYLIST_SONGS PS
JOIN SONGS S ON PS.SONG_ID = S.SONG_ID
GROUP BY PS.PLAYLIST_ID;
```

Find the duration of the longest song in an album:

```
SELECT ALBUM_ID, MAX(DURATION_IN_SECONDS) AS MAX_DURATION
FROM SONGS
WHERE ALBUM_ID IS NOT NULL
GROUP BY ALBUM_ID
ORDER BY ALBUM_ID;
```

SINGLE ROW FUNCTION

Find the ages of users:

```
SELECT NAME, EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM DATE_OF_BIRTH) AS AGE FROM
USERS;
```

Count how many songs a genre has:

```
SELECT GENRE, COUNT(*) AS SONG_COUNT FROM SONGS GROUP BY GENRE;
```

Calculate the average rating of songs from each artist:

```
SELECT S.ARTIST_ID, AVG(R.RATING) AS AVERAGE_RATING FROM SONGS S INNER JOIN USER_REVIEWS R ON
S.SONG_ID = R.SONG_ID GROUP BY ARTIST_ID;
```

VIEWS

Creates a view that shows the total number of followers for each artist:

```
CREATE VIEW ARTIST_FOLLOWER_COUNT AS
SELECT A.ARTIST_NAME, COUNT(F.FOLLOWER_ID) AS FOLLOWER_COUNT
FROM ARTISTS A LEFT JOIN FOLLOWERS F ON A.ARTIST_ID = F.ARTIST_ID
GROUP BY A.ARTIST_NAME;


SELECT * FROM ARTIST_FOLLOWER_COUNT;
```

SCHEMA OF ORGANIZATION

CREATE TABLE USERS (USER_ID VARCHAR2(20), NAME VARCHAR2(50) NOT NULL, USER_NAME VARCHAR2(50) NOT NULL UNIQUE, EMAIL VARCHAR2(20) NOT NULL UNIQUE, DATE_OF_BIRTH DATE, CONSTRAINT USERS_PK PRIMARY KEY(USER_ID));

INSERT INTO USERS VALUES ('FNW-100', 'Fatima Nadeem', 'Fatima_NW', 'FATIMANW', '03/13/2003');
INSERT INTO USERS VALUES ('TQZ-101', 'Tariq Qamar', 'Tariq_Q', 'TARIQQ', '10/12/1980');
INSERT INTO USERS VALUES ('RJL-102', 'Remus Lupin', 'Moony', 'REMUSL', '03/10/1960');
INSERT INTO USERS VALUES ('AK-103', 'Aamna Khalid', 'aamna_kayy', 'AAMNAK', '05/19/2004');
INSERT INTO USERS VALUES ('AS-104', 'Aisha Shahzad', 'Penguin', 'AISHASH', '06/03/2002');
INSERT INTO USERS VALUES ('MA-105', 'Muqaddas Ali', 'Muqs', 'MUQSALI', '12/17/2000');
INSERT INTO USERS VALUES ('SB-106', 'Sirius Black', 'Padfoot', 'SIRIUSB', '11/03/1959');
INSERT INTO USERS VALUES ('AD-107', 'Albus Dumbledore', 'Dumbly', 'DUMBLEDORE', '08/15/1881');
INSERT INTO USERS VALUES ('KK-108', 'Kvothe Kingkiller', 'Kvothe_the_Bloodless', 'KVOTHEK', NULL);
INSERT INTO USERS VALUES ('CH-109', 'Charlotte', 'Charlotte', 'CHARLOTTE', '07/11/1995');
INSERT INTO USERS VALUES ('HH-110', 'Hans Hubermann', 'Hans_H', 'HANS', '09/04/1950');
INSERT INTO USERS VALUES ('OMA-111', 'Olivier Armstrong', 'Northern_Wall_of_Briggs', 'GENERALARMSTRONG', '06/25/1987');
INSERT INTO USERS VALUES ('IR-112', 'Iroh', 'Dragon_of_the_West', 'IROH', '02/02/1952');
INSERT INTO USERS VALUES ('OZ-113', 'Ozai', 'Fire_Lord', 'OZAI', '11/20/1965');
INSERT INTO USERS VALUES ('AMC-114', 'Annie Cresta', 'Annieee', 'AC', '09/09/1996');
INSERT INTO USERS VALUES ('JJ-115', 'Julian Jade', 'Julian', 'JULIANJ', '03/08/1992');
INSERT INTO USERS VALUES ('WL-116', 'Wolfstar Lestrange', 'Wolf', 'WOLFSTAR', '07/17/1991');
INSERT INTO USERS VALUES ('PR-117', 'Robot IV', 'Prince_IV', 'ROBOTIV', NULL);
INSERT INTO USERS VALUES ('CG-118', 'Cardan Greenbriar', 'marriedtojude', 'KINGCARDAN', '09/13/2001');
INSERT INTO USERS VALUES ('EH-119', 'Elijah Hewson', 'elijahhewson', 'ELIJAHINHALER', '08/17/1999');

CREATE TABLE GENRES (GENRE_NAME VARCHAR2(20), GENRE_DESCRIPTION VARCHAR2(1000) UNIQUE, CONSTRAINT GENRES_PK PRIMARY KEY(GENRE_NAME));

INSERT INTO GENRES VALUES ('Pop', 'Popular music. Usually include repeated choruses and a catchy melody');
INSERT INTO GENRES VALUES ('Rock', 'Features guitar-based melodies and loud, powerful sound');
INSERT INTO GENRES VALUES ('Hip hop', 'Combines rapping, disc jockeying and beatboxing');
INSERT INTO GENRES VALUES ('Jazz', 'Features a mix of complex rhythms and tempos');
INSERT INTO GENRES VALUES ('Country', 'Features lyrics about love, heartbreak, nature, and small-town life');
INSERT INTO GENRES VALUES ('Classical', 'An ancient style of music known for its elaborate compositions and intricate harmonies');
INSERT INTO GENRES VALUES ('Heavy metal', 'Features loud, distorted electric guitars, heavy drums, powerful vocals and often dark, aggressive themes');
INSERT INTO GENRES VALUES ('Funk', 'Combining elements of soul and jazz');
INSERT INTO GENRES VALUES ('Indie', 'Often associated with creative expression and independent record labels');
INSERT INTO GENRES VALUES ('R&B', 'Characterized by sorrowful lyrics and bluesy melodies');
INSERT INTO GENRES VALUES ('Gospel', 'Characterized by its spiritual and uplifting messages, expressed through soulful melodies and powerful vocals')
INSERT INTO GENRES VALUES ('Alternative', 'Characterized by unconventional sound and approach to songwriting, usually experimental and not mainstream');
INSERT INTO GENRES VALUES ('Electronic dance', 'Features high-energy rhythm, strong bass lines, and use of other electronic instruments and effects');
INSERT INTO GENRES VALUES ('Nu metal', 'Combines the aggressive instrumentation of heavy metal characterized by its raw intensity, heavy riffs, and rap-infused vocals');

INSERT INTO GENRES VALUES ('Indie Folk', 'Characterized by intimate, acoustic-driven compositions with introspective lyrics, blending elements of indie rock and folk music');
INSERT INTO GENRES VALUES ('Soul', 'Characterized by bluesy vocals, rhythm and blues influence, and an emphasis on emotional intensity and social commentary')
INSERT INTO GENRES VALUES ('Ballad', 'Characterized by emotionally-driven songs featuring heartfelt lyrics and a focus on storytelling, conveying themes of love, heartbreak, and introspection');

CREATE TABLE MUSIC_LABELS (LABEL_ID VARCHAR2(10), LABEL_NAME VARCHAR2(50) NOT NULL UNIQUE, FOUNDER VARCHAR2(50) NOT NULL, YEAR_FOUNDED NUMBER(4) NOT NULL CHECK (YEAR_FOUNDED BETWEEN 1800 AND 2023), CONSTRAINT MUSIC_LABELS_PK PRIMARY KEY(LABEL_ID));
INSERT INTO MUSIC_LABELS VALUES ('LABEL001', 'Interscope Records', 'Jimmy Lovine', 1989);
INSERT INTO MUSIC_LABELS VALUES ('LABEL002', 'Columbia Records', 'Edward D. Easton', 1888);
INSERT INTO MUSIC_LABELS VALUES ('LABEL003', 'Big Hit Entertainment', 'Bang Si-hyuk', 2005);
INSERT INTO MUSIC_LABELS VALUES ('LABEL004', 'Republic Records', 'Monte Lipman', 1995);
INSERT INTO MUSIC_LABELS VALUES ('LABEL005', 'Atlantic Records', 'Ahmet Ertegun', 1947);
INSERT INTO MUSIC_LABELS VALUES ('LABEL006', 'Parkwood Entertainment', 'Beyoncé Knowles-Carter', 2008);
INSERT INTO MUSIC_LABELS VALUES ('LABEL007', 'Syco Music', 'Simon Cowell', 2002);
INSERT INTO MUSIC_LABELS VALUES ('LABEL008', 'XL Recordings', 'Richard Russell', 1989);
INSERT INTO MUSIC_LABELS VALUES ('LABEL009', 'Geffen Records', 'David Geffen', 1980);
INSERT INTO MUSIC_LABELS VALUES ('LABEL010', 'RCA Records', 'Eldridge Johnson', 1901);
INSERT INTO MUSIC_LABELS VALUES ('LABEL012', 'Polydor Records', 'Gustav Adolf Brachhausen', 1954);
INSERT INTO MUSIC_LABELS VALUES ('LABEL014', 'Sony Music Entertainment', 'Louis G. Sylvester', 1929);
INSERT INTO MUSIC_LABELS VALUES ('LABEL016', 'Brilliance Records', 'Ruben Nesse', 2018);
INSERT INTO MUSIC_LABELS VALUES ('LABEL017', 'Warner Bros. Records', 'Jack Warner', 1958);
INSERT INTO MUSIC_LABELS VALUES ('LABEL018', 'Virgin EMI Records', 'Richard Branson', 1931);

CREATE TABLE ARTISTS (ARTIST_ID VARCHAR2(10), ARTIST_NAME VARCHAR2(40) NOT NULL, MAIN_GENRE VARCHAR2(20), ARTIST_BIO VARCHAR2(1000), LABEL_ID VARCHAR(10), CONSTRAINT ARTISTS_PK PRIMARY KEY(ARTIST_ID), CONSTRAINT ARTISTS_GENRES_FK FOREIGN KEY(MAIN_GENRE) REFERENCES GENRES(GENRE_NAME), CONSTRAINT ARTISTS_MUSIC_LABELS FOREIGN KEY(LABEL_ID) REFERENCES MUSIC_LABELS(LABEL_ID));
INSERT INTO ARTISTS VALUES ('LDR-1985', 'Lana Del Rey', 'Indie', 'An enigmatic and ethereal songstress whose haunting vocals and dreamy cinematic soundscapes have redefined modern pop music, enchanting fans with her signature melancholic allure.', 'LABEL001');
INSERT INTO ARTISTS VALUES ('TS-1989', 'Taylor Swift', 'Country', 'The multi-talented singer-songwriter and global icon whose heartfelt lyrics and genre-spanning music have made her one of the most influential artists of our time, capturing the essence of love, growth, and personal empowerment.', 'LABEL004');
INSERT INTO ARTISTS VALUES ('ES-1991', 'Ed Sheeran', 'Pop', 'The British singer-songwriter extraordinaire, known for his soul-stirring lyrics, heartfelt melodies, and versatile musicality, captivating audiences worldwide with his raw authenticity and relatable storytelling.', 'LABEL005');
INSERT INTO ARTISTS VALUES ('B-1981', 'Beyoncé', 'Hip hop', 'The iconic Grammy-winning artist, whose unparalleled vocal range, mesmerizing performances, and empowering anthems have solidified her status as a global music legend, inspiring generations with her unmatched talent and undeniable charisma.', 'LABEL006');
INSERT INTO ARTISTS VALUES ('HS-1994', 'Harry Styles', 'Pop','A charismatic and boundary-pushing artist whose infectious charm, soulful voice, and fearless fashion sense have solidified him as a true icon of individuality, captivating audiences with his ever-evolving artistry.', 'LABEL002');
INSERT INTO ARTISTS VALUES ('LT-1991', 'Louis Tomlinson', 'Pop', 'The talented singer-songwriter and former member of One Direction, whose powerful vocals and heartfelt lyrics have showcased his ability to connect deeply with fans, leaving an indelible mark on the music industry.', 'LABEL007');
INSERT INTO ARTISTS VALUES ('IN-2013', 'Inhaler', 'Rock', 'A Dublin-based indie rock band known for their infectious melodies and poetic lyrics, creating a captivating sonic journey that resonates with listeners around the globe. Members include Elijah Hewson, Ryan McMahon, Robert Keating and Josh Jenkinson.', 'LABEL012');
INSERT INTO ARTISTS VALUES ('1D-2010', 'One Direction', 'Pop', 'A British-Irish boy band that stole hearts and took the world by storm with their catchy tunes and irresistible charm. Members include Harry Styles, Louis Tomlinson, Zayn Malik, Niall Horan and Liam Payne.', 'LABEL013');

INSERT INTO ARTISTS VALUES ('MA-2016', 'Maneskin', 'Rock', 'An Italian rock band that brought a fierce and electrifying energy to the music scene, captivating audiences with their unique style and powerful performances. Members include Damiano David, Victoria De Angelis, Thomas Raggi and Ethan Torchio.', 'LABEL014');
INSERT INTO ARTISTS VALUES ('BTS-2010', 'BTS', 'Electronic dance', 'The unstoppable force of South Korean music, combining stunning visuals, electrifying performances, and an ARMY of passionate fans worldwide. Members include RM, Jin, Suga, J-Hope, Jimin, V, and Jungkook.', 'LABEL003');
INSERT INTO ARTISTS VALUES ('AQ-1988', 'Adele', 'Soul', 'A powerhouse vocalist and songwriter whose emotionally charged ballads and soulful voice have earned her critical acclaim and global success.', 'LABEL008');
INSERT INTO ARTISTS VALUES ('CG-1998', 'Conan Gray', 'Pop', 'A rising pop artist known for his introspective lyrics, captivating vocals, and relatable storytelling that resonates with listeners.', 'LABEL015');
INSERT INTO ARTISTS VALUES ('NA-1991', 'Novo Amor', 'Indie Folk', 'An indie folk artist known for his ethereal soundscapes, delicate melodies, and emotive lyrics that create a deeply atmospheric musical experience.', 'LABEL016');
INSERT INTO ARTISTS VALUES ('LP-1996', 'Linkin Park', 'Nu metal', 'Linkin Park was an influential rock band known for their fusion of rock, alternative, and electronic elements, delivering powerful anthems that touched the hearts of millions.', 'LABEL017');
INSERT INTO ARTISTS VALUES ('TV-2012', 'The Vamps', 'Pop', 'A British pop band recognized for their catchy hooks, infectious energy, and vibrant performances that have amassed a dedicated fanbase worldwide.', 'LABEL018');
INSERT INTO ARTISTS VALUES ('OR-2003', 'Olivia Rodrigo', 'Pop', 'A breakout pop star celebrated for her raw and honest songwriting, resonating deeply with a generation through her heartfelt lyrics and emotive vocal delivery.', 'LABEL009');
INSERT INTO ARTISTS VALUES ('ZM-1993', 'Zayn Malik', 'R&B', 'The British heartthrob who captivated millions with his soulful voice, irresistible charisma, and striking good looks as a former member of One Direction.', 'LABEL010');
INSERT INTO ARTISTS VALUES ('JK-1997', 'Jungkook', 'Classical', 'The multi-talented golden maknae of BTS, whose powerful vocals, impressive dance skills, and adorable charm make hearts flutter worldwide.', 'LABEL011');

CREATE TABLE ALBUMS (ALBUM_ID VARCHAR2(10), ALBUM_TITLE VARCHAR2(40) NOT NULL, ARTIST_ID VARCHAR2(10) NOT NULL, RELEASE_DATE DATE NOT NULL, CONSTRAINT ALBUMS_PK PRIMARY KEY(ALBUM_ID), CONSTRAINT ALBUMS_ARTISTS_FK FOREIGN KEY(ARTIST_ID) REFERENCES ARTISTS(ARTIST_ID));
INSERT INTO ALBUMS VALUES ('BTD-2012', 'Born To Die', 'LDR-1985', '01/27/2012');
INSERT INTO ALBUMS VALUES ('EVR-2020', 'Evermore', 'TS-1989', '12/11/2020')
INSERT INTO ALBUMS VALUES ('FLK-2020', 'Folklore', 'TS-1989', '07/24/2020')
INSERT INTO ALBUMS VALUES ('DVD-2017', 'Divide', 'ES-1991', '03/03/2017')
INSERT INTO ALBUMS VALUES ('RNS-2022', 'Renaissance', 'B-1981', '07/29/2022');
INSERT INTO ALBUMS VALUES ('AQ-2015', '25', 'AQ-1988', '11/20/2015');
INSERT INTO ALBUMS VALUES ('MOS-2019', 'Map of the Soul: Persona', 'BTS-2010', '04/12/2019');
INSERT INTO ALBUMS VALUES ('SR-2021', 'Sour', 'OR-2003', '05/21/2021');
INSERT INTO ALBUMS VALUES ('HRH-2022', 'Harry''s House', 'HS-1994', '05/20/2022');
INSERT INTO ALBUMS VALUES ('LYT-2018', 'Love Yourself: Tears', 'BTS-2010', '05/18/2018');
INSERT INTO ALBUMS VALUES ('NIL-2021', 'Nobody Is Listening', 'ZM-1993', '01/15/2021');
INSERT INTO ALBUMS VALUES ('FTF-2022', 'Faith in the Future', 'LT-1991', '11/11/2022');
INSERT INTO ALBUMS VALUES ('C&B-2023', 'Cuts & Bruises', 'IN-2013', '02/17/2023');
INSERT INTO ALBUMS VALUES ('IWA-2021', 'It Won''t Always Be Like This', 'IN-2013', '07/09/2023');
INSERT INTO ALBUMS VALUES ('TMH-2012', 'Take Me Home', '1D-2010', '11/12/2012');
INSERT INTO ALBUMS VALUES ('SA-2022', 'Superache', 'CG-1998', '06/24/2022');
INSERT INTO ALBUMS VALUES ('KK-2020', 'Kid Krow', 'CG-1998', '03/20/2020');

CREATE TABLE SONGS (SONG_ID NUMBER(4), SONG_TITLE VARCHAR2(40) NOT NULL, ARTIST_ID VARCHAR2(10) NOT NULL, ALBUM_ID VARCHAR2(10), GENRE VARCHAR2(20), DURATION_IN_SECONDS NUMBER(5) CHECK (DURATION_IN_SECONDS >= 0), RELEASE_DATE DATE, CONSTRAINT SONGS_PK PRIMARY KEY(SONG_ID), CONSTRAINT SONGS_ARTISTS_FK FOREIGN KEY(ARTIST_ID) REFERENCES ARTISTS(ARTIST_ID), CONSTRAINT SONGS_ALBUMS_FK FOREIGN KEY(ALBUM_ID) REFERENCES ALBUMS(ALBUM_ID), CONSTRAINT SONGS_GENRES_FK FOREIGN KEY(GENRE) REFERENCES GENRES(GENRE_NAME));

INSERT INTO SONGS VALUES (1505, 'National Anthem', 'LDR-1985', 'BTD-2012', 'Pop', 231, '05/15/2012');
INSERT INTO SONGS VALUES (2701 , 'Radio', 'LDR-1985', 'BTD-2012', 'Indie', 217, '01/27/2012');
INSERT INTO SONGS VALUES (1112, 'Willow', 'TS-1989', 'EVR-2020', 'Country', 214, '12/11/2020');
INSERT INTO SONGS VALUES (1005 , 'Young and Beautiful', 'LDR-1985', NULL, 'Indie', 239, '05/10/2013');
INSERT INTO SONGS VALUES (0504, 'Carry You', 'NA-1991', NULL, 'Indie Folk', 274, '04/05/2017');
INSERT INTO SONGS VALUES (0303, 'Perfect', 'ES-1991', 'DVD-2017', 'Pop', 280, '03/03/2017');
INSERT INTO SONGS VALUES (3012, 'Born to Die', 'LDR-1985', 'BTD-2012', 'Indie' , 287, '12/30/2011');
INSERT INTO SONGS VALUES (1007, 'Spirit', 'B-1981', NULL, 'Gospel', 271, '07/10/2019');
INSERT INTO SONGS VALUES (1805, 'Fake Love', 'BTS-2010', 'LYT-2018' , 'Hip hop', 319 , '05/18/2018');
INSERT INTO SONGS VALUES (2407, 'Seven', 'TS-1989', 'FLK-2020', 'Indie Folk', 208 , '07/24/2020');
INSERT INTO SONGS VALUES (0809, 'Numb', 'LP-1996', NULL, 'Nu metal', 188, '09/08/2003');
INSERT INTO SONGS VALUES (0801, 'Vibez', 'ZM-1993','NIL-2021', 'R&B', 163, '01/08/2021');
INSERT INTO SONGS VALUES (2107, 'Back To You', 'LT-1991', NULL, 'Pop', 190, '07/21/2017');
INSERT INTO SONGS VALUES (1705, 'My Honest Face', 'IN-2013','IWA-2021', 'Rock', 273, '05/17/2019');
INSERT INTO SONGS VALUES (1702, 'Perfect Storm', 'IN-2013','C&B-2023', 'Alternative', 223, '02/17/2023');
INSERT INTO SONGS VALUES (2911, 'Rolling in the Deep', 'AQ-1988', NULL, 'Soul', 234 , '11/29/2010');
INSERT INTO SONGS VALUES (2709, 'Beggin'", 'MA-2016', NULL, 'Rock', 211, '09/27/2021')
INSERT INTO SONGS VALUES (2005, 'Matilda', 'HS-1994','HRH-2022', 'Ballad', 245, '05/20/2022');
INSERT INTO SONGS VALUES (2406, 'The Exit', 'CG-1998','SA-2022', 'Pop', 221, '06/24/2022');
INSERT INTO SONGS VALUES (0406, 'Still With You', 'JK-1997', NULL, 'Jazz', 239, '06/04/2020');
INSERT INTO SONGS VALUES (0309, 'Brutal', 'OR-2003', 'SR-2021', 'Alternative', 174, '09/03/2021');
INSERT INTO SONGS VALUES (2809,'Live While We"re Young', '1D-2010', 'TMH-2012', 'Pop', 200, '09/28/2012');
INSERT INTO SONGS VALUES (1905, 'Somebody To You', 'TV-2012', NULL, 'Alternative', 199, '05/19/2014');
INSERT INTO SONGS VALUES (1204, 'Boy With Luv','BTS-2010', 'MOS-2019', 'Funk', 253, '04/12/2019');
INSERT INTO SONGS VALUES (0910, 'In the End', 'LP-1996', NULL, 'Rock', 217, '10/09/2001');

CREATE TABLE PLAYLISTS (PLAYLIST_ID VARCHAR2(10), PLAYLIST_NAME VARCHAR2(40) NOT NULL, USER_ID VARCHAR2(20) NOT NULL, CREATION_DATE DATE, PLAYLIST_DESCRIPTION VARCHAR2(1000), CONSTRAINT PLAYLISTS_PK PRIMARY KEY(PLAYLIST_ID), CONSTRAINT PLAYLISTS_USERS_FK FOREIGN KEY(USER_ID) REFERENCES USERS(USER_ID));

INSERT INTO PLAYLISTS VALUES ('FN-2106', 'HER', 'FNW-100', '06/21/2023', 'Just Lana');
INSERT INTO PLAYLISTS VALUES ('FN-1212', 'PEACE', 'FNW-100', '12/12/2022', 'Songs to sleep to');
INSERT INTO PLAYLISTS VALUES ('TQ-1108' , 'RANDOM', 'TQZ-101', '08/11/2016', 'Random songs I like');
INSERT INTO PLAYLISTS VALUES ('AD-2404', 'The Longevity Lullabies', 'AD-107', '04/24/2012', 'Songs that keep me spry and sly');
INSERT INTO PLAYLISTS VALUES ('AD-1403', 'Magical Melodies', 'AD-107', '03/14/2017', 'Trillalalalaaa');
INSERT INTO PLAYLISTS VALUES ('SB-1103', 'Rebel"s Rhapsodies', 'SB-106', '03/11/2023', 'LOUDER');
INSERT INTO PLAYLISTS VALUES ('RJL-1206', 'Whispers of the Moon', 'RJL-102', '06/12/2020', 'For moonlit strolls');
INSERT INTO PLAYLISTS VALUES ('RJL-1003', 'Echoes of Solitude', 'RJL-102', '03/10/2014', '</3');
INSERT INTO PLAYLISTS VALUES ('KK-3101', 'Songs of the Wind', 'KK-108', '10/31/2014', NULL);

INSERT INTO PLAYLISTS VALUES ('IR-2503', 'Poetic Tea Leaves', 'IR-112', '03/25/2015', 'Tea-time melodies');
INSERT INTO PLAYLISTS VALUES ('AK-1506', 'let me cry in peace', 'AK-103', '06/15/2022', 'i"m in love with these songs');
INSERT INTO PLAYLISTS VALUES ('AK-0105', 'i"d die for you', 'AK-103', '05/01/2021', 'dedicated to my besties <3');
INSERT INTO PLAYLISTS VALUES ('OMA-1109', 'Unyielding Beats', 'OMA-111', '09/11/2016', NULL);
INSERT INTO PLAYLISTS VALUES ('MQ-1212', 'ARMY', 'MA-105', '12/12/2022', 'BTS stan');
INSERT INTO PLAYLISTS VALUES ('AS-0101', 'Waddle Jams', 'AS-104', '01/01/2019', 'Dancing on Ice');

CREATE TABLE PLAYLIST_SONGS (PLAYLIST_ID VARCHAR2(10), SONG_ID NUMBER(4), SONG_POSITION NUMBER(4), CONSTRAINT PLAYLIST_SONGS_CK PRIMARY KEY(PLAYLIST_ID, SONG_ID), CONSTRAINT PLAYLIST_SONGS_PLAYLISTS FOREIGN KEY(PLAYLIST_ID) REFERENCES PLAYLISTS(PLAYLIST_ID), CONSTRAINT PLAYLIST_SONGS_SONGS FOREIGN KEY(SONG_ID) REFERENCES SONGS(SONG_ID));
INSERT INTO PLAYLIST_SONGS VALUES ('FN-2106', 2701, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('FN-2106', 1505, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('FN-2106', 3012, 3);
INSERT INTO PLAYLIST_SONGS VALUES ('FN-1212', 3012, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('TQ-1108', 1112, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('TQ-1108', 0303, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('AD-2404', 1005, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('AD-2404', 2809, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('AD-1403', 1112, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('AD-1403', 1007, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('SB-1103', 0910, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('SB-1103', 0809, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('SB-1103', 2709, 3);
INSERT INTO PLAYLIST_SONGS VALUES ('RJL-1206', 0504, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('RJL-1003', 0309, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('RJL-1003', 2005, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('KK-3101', 2911, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('IR-2503', 1805, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('IR-2503', 2406, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('AK-1506', 504, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('AK-1506', 2005, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('AK-1506', 2407, 3);
INSERT INTO PLAYLIST_SONGS VALUES ('AK-0105', 1702, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('AK-0105', 2709, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('OMA-1109', 1007, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('MQ-1212', 1805, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('MQ-1212', 1204, 2);
INSERT INTO PLAYLIST_SONGS VALUES ('MQ-1212', 0406, 3);
INSERT INTO PLAYLIST_SONGS VALUES ('AS-0101', 0303, 1);
INSERT INTO PLAYLIST_SONGS VALUES ('AS-0101', 1905, 2);

CREATE TABLE USER_LIBRARY (LIBRARY_ID VARCHAR2(10) NOT NULL, USER_ID VARCHAR2(20) NOT NULL, ALBUM_ID VARCHAR2(10), ARTIST_ID VARCHAR2(10), LIBRARY_POSITION_ID VARCHAR2(20), DATE_ADDED DATE, CONSTRAINT USER_LIBRARY_PK PRIMARY KEY(LIBRARY_POSITION_ID), CONSTRAINT USER_LIBRARY_USERS_FK FOREIGN KEY(USER_ID) REFERENCES USERS(USER_ID), CONSTRAINT USER_LIBRARY_ALBUMS_FK FOREIGN KEY(ALBUM_ID) REFERENCES ALBUMS(ALBUM_ID), CONSTRAINT USER_LIBRARY_ARTISTS_FK FOREIGN KEY(ARTIST_ID) REFERENCES ARTISTS(ARTIST_ID));

INSERT INTO USER_LIBRARY VALUES ('LIB-001', 'FNW-100', 'BTD-2012', NULL,'FN-2806', '06/28/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-001', 'FNW-100', NULL, 'LDR-1985' ,'FN-0501', '01/05/2022');
INSERT INTO USER_LIBRARY VALUES ('LIB-001', 'FNW-100', 'RNS-2022', NULL,'FN-1309', '09/13/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-002', 'TQZ-101', 'FLK-2020', NULL,'TQ-0511','11/05/2016');
INSERT INTO USER_LIBRARY VALUES ('LIB-002', 'TQZ-101', NULL, 'TS-1989' ,'TQ-3110', '10/31/2018');
INSERT INTO USER_LIBRARY VALUES ('LIB-002', 'TQZ-101', 'EVR-2020', NULL ,'TQ-0603', '03/06/2019');
INSERT INTO USER_LIBRARY VALUES ('LIB-003', 'AS-104', NULL, 'LP-1996' ,'AS-1611', '11/16/2022');
INSERT INTO USER_LIBRARY VALUES ('LIB-003', 'AS-104', NULL, 'NA-1991' ,'AS-1605', '05/16/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-003', 'AS-104', 'MOS-2019', NULL ,'AS-1909', '09/19/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-003', 'AS-104', 'KK-2020', NULL ,'AS-2310', '10/23/2018');
INSERT INTO USER_LIBRARY VALUES ('LIB-003', 'AS-104', 'DVD-2017', NULL ,'AS-2402', '02/24/2015');
INSERT INTO USER_LIBRARY VALUES ('LIB-004', 'AK-103', NULL, 'IN-2013' ,'AK-1701', '01/17/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-004', 'AK-103', NULL, 'CG-1998' ,'AK-1905', '05/19/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-004', 'AK-103', 'SA-2022', NULL ,'AK-1205', '05/12/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-004', 'AK-103', 'HRH-2022', NULL ,'AK-2704', '04/27/2019');
INSERT INTO USER_LIBRARY VALUES ('LIB-005', 'MA-105', NULL, 'BTS-2010' ,'MA-1508', '08/15/2020');
INSERT INTO USER_LIBRARY VALUES ('LIB-005', 'MA-105', 'LYT-2018', NULL ,'MA-1104', '04/11/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-005', 'MA-105', 'MOS-2019', NULL ,'MA-0707', '07/07/2019');
INSERT INTO USER_LIBRARY VALUES ('LIB-006', 'OMA-111', NULL, 'B-1981' ,'OMA-2211', '11/22/2016');
INSERT INTO USER_LIBRARY VALUES ('LIB-006', 'OMA-111', 'RNS-2022', NULL ,'OMA-2908', '08/29/2013');
INSERT INTO USER_LIBRARY VALUES ('LIB-007', 'RJL-102', NULL,'1D-2010' ,'RJL-1209', '09/12/2015');
INSERT INTO USER_LIBRARY VALUES ('LIB-007', 'RJL-102', 'TMH-2012', NULL ,'RJL-1702', '02/17/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-008', 'HH-110', 'C&B-2023', NULL ,'HH-1210', '10/12/2022');
INSERT INTO USER_LIBRARY VALUES ('LIB-008', 'HH-110', NULL, 'ZM-1993' ,'HH-1605', '05/16/2016');
INSERT INTO USER_LIBRARY VALUES ('LIB-009', 'KK-108', 'AQ-2015', NULL ,'KK-1903', '03/19/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-010', 'IR-112', 'LYT-2018', NULL ,'IR-1806', '06/18/2011');
INSERT INTO USER_LIBRARY VALUES ('LIB-010', 'IR-112', NULL, 'AQ-1988' ,'IR-1110', '11/10/2010');
INSERT INTO USER_LIBRARY VALUES ('LIB-011', 'EH-119', 'KK-2020', NULL ,'EH-0504', '04/05/2018');
INSERT INTO USER_LIBRARY VALUES ('LIB-012', 'CG-118', NULL, 'MA-2016' ,'CG-1711', '11/17/2019');
INSERT INTO USER_LIBRARY VALUES ('LIB-013', 'JJ-115', 'SR-2021', NULL ,'JJ-1407', '07/14/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-014', 'AMC-114', NULL, 'TV-2012' ,'AMC-0112', '12/01/2014');
INSERT INTO USER_LIBRARY VALUES ('LIB-014', 'AMC-114', NULL, 'ES-1991' ,'AMC-2504', '04/25/2017');
INSERT INTO USER_LIBRARY VALUES ('LIB-015', 'CH-109', 'IWA-2021', NULL ,'CH-2601', '01/26/2015');
INSERT INTO USER_LIBRARY VALUES ('LIB-015', 'CH-109', 'FTF-2022', NULL ,'CH-2207', '07/22/2022');
INSERT INTO USER_LIBRARY VALUES ('LIB-015', 'CH-109', NULL, 'JK-1997' ,'CH-2302', '02/23/2023');
INSERT INTO USER_LIBRARY VALUES ('LIB-016', 'WL-116', 'DVD-2017', NULL ,'WL-1610', '10/16/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-016', 'WL-116', 'FLK-2020', NULL ,'WL-0311', '11/03/2021');
INSERT INTO USER_LIBRARY VALUES ('LIB-017', 'AD-107', NULL, 'B-1981' ,'AD-0911', '11/09/2012');

CREATE TABLE PLAYBACK_HISTORY (PLAYBACK_ID NUMBER(3), USER_ID VARCHAR2(20) NOT NULL, SONG_ID NUMBER(4) NOT NULL, PLAYLIST_ID VARCHAR2(10), LISTENED_DATE DATE NOT NULL, CONSTRAINT PLAYBACK_HISTORY_PK PRIMARY KEY(PLAYBACK_ID), CONSTRAINT PLAYBACK_HISTORY_USERS FOREIGN KEY(USER_ID) REFERENCES USERS(USER_ID), CONSTRAINT PLAYBACK_HISTORY_SONGS FOREIGN KEY(SONG_ID) REFERENCES SONGS(SONG_ID), CONSTRAINT PLAYBACK_HISTORY_PLAYLISTS FOREIGN KEY(PLAYLIST_ID) REFERENCES PLAYLISTS(PLAYLIST_ID));

INSERT INTO PLAYBACK_HISTORY VALUES (0, 'TQZ-101', 1112 , 'TQ-1108', '07/07/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (1, 'JJ-115', 0801 , NULL, '07/07/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (2, 'TQZ-101', 0303, 'TQ-1108', '07/06/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (3, 'FNW-100', 1505, 'FN-2106', '07/06/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (4, 'RJL-102', 504, 'RJL-1206', '07/06/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (5, 'FNW-100', 3012, 'FN-2106', '07/05/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (6, 'FNW-100', 0406, NULL, '07/04/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (7, 'AD-107', 2809, 'AD-2404', '07/03/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (8, 'AK-103', 1702, 'AK-0105', '07/02/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (9, 'IR-112', 1905, NULL, '07/01/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (10, 'OMA-111', 1007, 'OMA-1109', '06/30/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (11, 'MA-105', 1204, 'MQ-1212', '06/30/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (12, 'RJL-102', 2005, 'RJL-1003', '06/28/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (13, 'CH-109', 2911, NULL, '06/26/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (14, 'WL-116', 1005, NULL, '06/25/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (15, 'KK-108', 406, NULL, '06/25/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (16, 'AD-107', 1007, NULL, '06/24/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (17, 'SB-106', 0809, 'SB-1103', '06/23/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (18, 'SB-106', 2709, 'SB-1103', '06/20/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (19, 'HH-110', 2407, NULL, '06/18/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (20, 'AS-104', 303, 'AS-0101', '06/15/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (21, 'OMA-111', 1805, NULL, '06/15/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (22, 'IR-112', 1005, NULL, '06/15/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (22, 'OZ-113', 0309, NULL, '06/14/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (23, 'TQZ-101', 2406, 'TQ-1108', '06/12/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (24, 'TQZ-101', 2911, 'TQ-1108', '06/11/2023');
INSERT INTO PLAYBACK_HISTORY VALUES (25, 'AMC-114', 2701, NULL, '06/11/2023');

CREATE TABLE RECOMMENDATIONS (RECOMMENDATION_ID NUMBER(3), RECOMMENDED_SONG_ID NUMBER(4) NOT NULL, PLAYLIST_ID VARCHAR(10) NOT NULL, CONSTRAINT RECOMMENDATIONS_PK PRIMARY KEY(RECOMMENDATION_ID), CONSTRAINT RECS_SONGS_FK FOREIGN KEY(RECOMMENDED_SONG_ID) REFERENCES SONGS(SONG_ID), CONSTRAINT RECS_PLAYLISTS_FK FOREIGN KEY(PLAYLIST_ID) REFERENCES PLAYLISTS(PLAYLIST_ID));

INSERT INTO RECOMMENDATIONS VALUES (101, 1005, 'FN-2106');
INSERT INTO RECOMMENDATIONS VALUES (102, 2406, 'FN-2106');
INSERT INTO RECOMMENDATIONS VALUES (103, 2005, 'FN-1212');
INSERT INTO RECOMMENDATIONS VALUES (104, 2701, 'FN-1212');
INSERT INTO RECOMMENDATIONS VALUES (105, 2911, 'TQ-1108');

INSERT INTO RECOMMENDATIONS VALUES (106, 0801, 'TQ-1108');
INSERT INTO RECOMMENDATIONS VALUES (107, 3012, 'AD-2404');
INSERT INTO RECOMMENDATIONS VALUES (108, 1505, 'AD-2404');
INSERT INTO RECOMMENDATIONS VALUES (109, 2407, 'AD-1403');
INSERT INTO RECOMMENDATIONS VALUES (110, 0303, 'AD-1403');
INSERT INTO RECOMMENDATIONS VALUES (111, 1702, 'SB-1103');
INSERT INTO RECOMMENDATIONS VALUES (112, 1705, 'SB-1103');
INSERT INTO RECOMMENDATIONS VALUES (113, 0910, 'RJL-1206');
INSERT INTO RECOMMENDATIONS VALUES (114, 0406, 'RJL-1206');
INSERT INTO RECOMMENDATIONS VALUES (115, 2005, 'RJL-1003');
INSERT INTO RECOMMENDATIONS VALUES (116, 3012, 'RJL-1003');
INSERT INTO RECOMMENDATIONS VALUES (117, 2701, 'KK-3101');
INSERT INTO RECOMMENDATIONS VALUES (118, 1007, 'KK-3101');
INSERT INTO RECOMMENDATIONS VALUES (119, 504, 'IR-2503');
INSERT INTO RECOMMENDATIONS VALUES (120, 1112, 'IR-2503');
INSERT INTO RECOMMENDATIONS VALUES (121, 406, 'AK-1506');
INSERT INTO RECOMMENDATIONS VALUES (122, 3012, 'AK-1506');
INSERT INTO RECOMMENDATIONS VALUES (123, 1705, 'AK-0105');
INSERT INTO RECOMMENDATIONS VALUES (124, 2911, 'AK-0105');
INSERT INTO RECOMMENDATIONS VALUES (125, 1702, 'OMA-1109');
INSERT INTO RECOMMENDATIONS VALUES (126, 1705, 'OMA-1109');
INSERT INTO RECOMMENDATIONS VALUES (127, 303, 'MQ-1212');
INSERT INTO RECOMMENDATIONS VALUES (128, 1505, 'MQ-1212');
INSERT INTO RECOMMENDATIONS VALUES (129, 1204, 'AS-0101');
INSERT INTO RECOMMENDATIONS VALUES (130, 2809, 'AS-0101');

CREATE TABLE SUBSCRIPTION_PLANS (PLAN_ID NUMBER(4), PLAN_NAME VARCHAR2(15) NOT NULL UNIQUE, MONTHLY_COST_IN_RS NUMBER(5) CHECK (MONTHLY_COST_IN_RS >= 0), PLAN_DESCRIPTION VARCHAR2(1000), CONSTRAINT SUBSCRIPTION_PLAN_PK PRIMARY KEY(PLAN_ID));
INSERT INTO SUBSCRIPTION_PLANS VALUES (0, 'Free', 0, 'Ads after each song, No skips');
INSERT INTO SUBSCRIPTION_PLANS VALUES (25, 'Basic', 300, 'One ad after five songs, No skips');
INSERT INTO SUBSCRIPTION_PLANS VALUES (50, 'Standard', 600, 'One ad per hour, Ten skips per day');
INSERT INTO SUBSCRIPTION_PLANS VALUES (100, 'Premium', 1200, 'No ads, Unlimited skips');

CREATE TABLE SUBSCRIPTIONS (SUBSCRIPTION_ID NUMBER(4), USER_ID VARCHAR(20) NOT NULL, PLAN_ID NUMBER(4) NOT NULL, START_DATE DATE NOT NULL, EXPIRATION_DATE DATE NOT NULL CHECK (EXPIRATION_DATE >= START_DATE), CONSTRAINT SUBSCRIPTIONS_PK PRIMARY KEY(SUBSCRIPTION_ID), CONSTRAINT SUBSCRIPTIONS_USERS_FK FOREIGN KEY(USER_ID) REFERENCES USERS(USER_ID), CONSTRAINT SUBS_SUBSCRIPTION_PLANS_FK FOREIGN KEY(PLAN_ID) REFERENCES SUBSCRIPTION_PLANS(PLAN_ID));
INSERT INTO SUBSCRIPTIONS VALUES (4001, 'FNW-100', 100, '07/07/2023', '08/06/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4002, 'TQZ-101', 50, '07/03/2023', '08/02/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4003, 'RJL-102', 0, '07/01/2023', '07/30/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4004, 'AMC-114', 0, '07/01/2023', '07/30/2023');

INSERT INTO SUBSCRIPTIONS VALUES (4005, 'HH-110', 0, '07/01/2023', '07/30/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4006, 'AS-104', 0, '07/01/2023', '07/30/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4007, 'AK-103', 100, '06/21/2023', '07/20/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4008, 'CG-118', 50, '06/19/2023', '07/18/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4009, 'KK-108', 25, '05/10/2023', '06/09/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4010, 'IR-112', 100, '05/01/2023', '05/31/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4011, 'WL-116', 50, '05/01/2023', '05/31/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4012, 'AD-107', 100, '04/25/2023', '05/24/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4013, 'SB-106', 100, '04/09/2023', '05/08/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4014, 'JJ-115', 50, '04/01/2023', '04/30/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4015, 'OMA-111', 25, '04/01/2023', '04/30/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4016, 'CH-109', 25, '03/28/2023', '04/27/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4017, 'MA-105', 50, '03/01/2023', '03/31/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4018, 'OZ-113', 100, '02/12/2023', '03/11/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4019, 'PR-117', 25, '02/01/2023', '02/28/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4020, 'EH-119', 100, '02/01/2023', '02/28/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4021, 'JJ-115', 50, '02/01/2023', '02/28/2023');
INSERT INTO SUBSCRIPTIONS VALUES (4022, 'AD-107', 25, '01/19/2023', '02/28/2023');

CREATE TABLE PAYMENTS (PAYMENT_ID NUMBER(4), SUBSCRIPTION_ID NUMBER(4) NOT NULL, PAYMENT_DUE_DATE DATE NOT NULL, PAYMENT_DATE DATE, AMOUNT_PAID NUMBER(4) CHECK (AMOUNT_PAID >= 0), PAYMENT_STATUS VARCHAR2(15) NOT NULL, CONSTRAINT PAYMENTS_PK PRIMARY KEY(PAYMENT_ID), CONSTRAINT PAYMENTS_SUBSCRIPTIONS_FK FOREIGN KEY(SUBSCRIPTION_ID) REFERENCES SUBSCRIPTIONS(SUBSCRIPTION_ID));
INSERT INTO PAYMENTS VALUES (8501, 4001, '07/17/2023', '07/09/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8502, 4002, '07/13/2023', NULL, NULL, 'Pending');
INSERT INTO PAYMENTS VALUES (8503, 4007, '07/01/2023', '06/22/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8504, 4008, '06/29/2022', NULL, NULL,'Pending');
INSERT INTO PAYMENTS VALUES (8505, 4009, '05/20/2023', '05/19/2023', 300, 'Processed');
INSERT INTO PAYMENTS VALUES (8506, 4010, '05/11/2023', '05/05/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8507, 4011, '05/11/2023', '05/03/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8508, 4012, '05/05/2023', '04/25/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8509, 4013, '04/19/2023', NULL, NULL, 'Pending');
INSERT INTO PAYMENTS VALUES (8510, 4014, '04/11/2023', '04/02/2023', 600, 'Processed');
INSERT INTO PAYMENTS VALUES (8511, 4015, '04/11/2023', '04/01/2023', 300, 'Processed');
INSERT INTO PAYMENTS VALUES (8512, 4016, '04/08/2023', NULL, NULL, 'Pending');
INSERT INTO PAYMENTS VALUES (8513, 4017, '03/11/2023', NULL, NULL, 'Pending');
INSERT INTO PAYMENTS VALUES (8514, 4018, '02/22/2023', '02/21/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8515, 4019, '02/11/2023', NULL, NULL, 'Pending');
INSERT INTO PAYMENTS VALUES (8516, 4020, '02/11/2023', '02/01/2023', 1200, 'Processed');
INSERT INTO PAYMENTS VALUES (8517, 4021, '02/11/2023', '02/05/2023', 600, 'Processed');
INSERT INTO PAYMENTS VALUES (8518, 4022, '01/29/2023', '01/20/2023', 300, 'Processed');

CREATE TABLE USER_REVIEWS (REVIEW_ID VARCHAR2(20), USER_ID VARCHAR2(20) NOT NULL, SONG_ID NUMBER(4) NOT NULL, RATING NUMBER(2,1) NOT NULL CHECK (RATING >= 0 AND RATING <= 5), COMMENTS VARCHAR2(1000), CONSTRAINT USER_REVIEWS_PK PRIMARY KEY(REVIEW_ID), CONSTRAINT USER_REVIEWS_USERS_FK FOREIGN KEY(USER_ID) REFERENCES USERS(USER_ID), CONSTRAINT USER_REVIEWS_SONGS_FK FOREIGN KEY(SONG_ID) REFERENCES SONGS(SONG_ID));

INSERT INTO USER_REVIEWS VALUES ('RV-001', 'FNW-100', 3012, 5, 'QUEEEEEEN!!!!!');
INSERT INTO USER_REVIEWS VALUES ('RV-002', 'AK-103', 1705, 5, 'ELIJAH <3');
INSERT INTO USER_REVIEWS VALUES ('RV-003', 'PR-117', 809, 3.5, 'My antenna feels numb');
INSERT INTO USER_REVIEWS VALUES ('RV-004', 'AD-107', 2701, 5, 'Ahh.. Music');
INSERT INTO USER_REVIEWS VALUES ('RV-005', 'OZ-113', 0309, 2, 'Not what I expected this to be about..');
INSERT INTO USER_REVIEWS VALUES ('RV-006', 'RJL-102', 2809, 4.5, NULL);
INSERT INTO USER_REVIEWS VALUES ('RV-007', 'WL-116', 0303, 5, '*Sobs*');
INSERT INTO USER_REVIEWS VALUES ('RV-008', 'AD-107', 1112, 5, 'Can''t stop humming this');
INSERT INTO USER_REVIEWS VALUES ('RV-009', 'SB-106', 1007, 4, 'SPIRIIIIIIIIIT');
INSERT INTO USER_REVIEWS VALUES ('RV-010', 'IR-112', 1805, 5, 'Very talented children');
INSERT INTO USER_REVIEWS VALUES ('RV-011', 'AMC-114', 0303, 3.5, NULL);
INSERT INTO USER_REVIEWS VALUES ('RV-012', 'FNW-100', 1005, 2.5, NULL);
INSERT INTO USER_REVIEWS VALUES ('RV-013', 'AD-107', 3012, 1, 'I was born to live, thank you very much');
INSERT INTO USER_REVIEWS VALUES ('RV-014', 'JJ-115', 801, 3, 'Didn''t give me the vibez');
INSERT INTO USER_REVIEWS VALUES ('RV-015', 'MA-105', 1805, 4, NULL);
INSERT INTO USER_REVIEWS VALUES ('RV-016', 'SB-106', 2005, 5, 'Wowch');
INSERT INTO USER_REVIEWS VALUES ('RV-017', 'CG-118', 2709, 5, 'Never gets old!');

CREATE TABLE MUSIC_CHARTS (CHART_ID VARCHAR2(10), CHART_NAME VARCHAR2(100) NOT NULL, CHART_TYPE VARCHAR2(20), GENRE VARCHAR2(20), CHART_DESCRIPTION VARCHAR2(1000), CREATION_DATE DATE NOT NULL DEFAULT SYSDATE, CONSTRAINT MUSIC_CHARTS_PK PRIMARY KEY(CHART_ID), CONSTRAINT MUSIC_CHARTS_GENRES_FK FOREIGN KEY(GENRE) REFERENCES GENRES(GENRE_NAME));

INSERT INTO MUSIC_CHARTS VALUES ('CH-001', 'Top 3 Hits!', 'Weekly', 'Pop', 'Top most popular songs of the week', '07/03/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-002', 'Rock Classics', 'All Time', 'Rock', 'The greatest rock classics of all time ', '06/14/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-003', 'K-pop Sensation', 'Monthly', NULL, 'K-Pop biggest hits of the week', '06/01/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-004', 'Double Hitters', 'Daily', NULL, 'Today''s Top 2', '05/25/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-005', 'R&B Groove', 'Monthly', 'R&B', 'The smoothest R&B track of the month', '08/01/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-006', 'Throwback Hits', 'All Time', NULL, 'Nostalgic hits from the 2000s', '07/05/2022');
INSERT INTO MUSIC_CHARTS VALUES ('CH-007', 'Indie Vibes', 'Weekly', 'Indie', 'Discover the best indie tracks of the week', '07/09/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-008', 'Funky Fiesta', 'Weekly', 'Funk', 'Get into the funky groove with the hottest track', '07/05/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-009', 'Year-End Top Two Hits', 'Yearly', NULL, 'The two most popular songs of the year', '12/31/2022');
INSERT INTO MUSIC_CHARTS VALUES ('CH-010', 'Trending Now', 'Daily', NULL, 'The song currently trending worldwide', '07/06/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-011', 'Cultural Crossover', 'Weekly', NULL, 'Discover the captivating crossroads of music, where diverse cultures collide', '06/06/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-012', 'Gospel Glory', 'Weekly', 'Gospel', 'Experience the uplifting power of Gospel music', '05/16/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-013', 'Alternative Odyssey', 'All Time', 'Alternative', 'Embark on a musical journey through the realms of best alternative sounds', '07/08/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-014', 'Metal Mayhem', 'Daily', 'Nu metal','Unleash the ultimate metal anthem that will ignite your soul with its blistering intensity', '03/13/2023');
INSERT INTO MUSIC_CHARTS VALUES ('CH-015', 'Folk Fusion', 'Weekly', 'Indie Folk', 'This week''s harmonious blend of two captivating songs that intertwine traditional folk roots with contemporary allure', '04/13/2023');

CREATE TABLE CHART_SONGS (CHART_ID VARCHAR2(10), SONG_ID NUMBER(4), RANK NUMBER(3) CHECK (RANK >= 1 AND RANK <= 100), CONSTRAINT CHART_SONGS_PK PRIMARY KEY(CHART_ID, SONG_ID), CONSTRAINT CHART_SONGS_MUSIC_CHARTS_FK FOREIGN KEY(CHART_ID) REFERENCES MUSIC_CHARTS(CHART_ID), CONSTRAINT CHART_SONGS_SONGS_FK FOREIGN KEY(SONG_ID) REFERENCES SONGS(SONG_ID));

INSERT INTO CHART_SONGS VALUES ('CH-001', 1505, 1);
INSERT INTO CHART_SONGS VALUES ('CH-001', 1112, 2);
INSERT INTO CHART_SONGS VALUES ('CH-001', 0303, 3);
INSERT INTO CHART_SONGS VALUES ('CH-002', 0910, 1);
INSERT INTO CHART_SONGS VALUES ('CH-002', 1705, 2);
INSERT INTO CHART_SONGS VALUES ('CH-002', 2709, 3);
INSERT INTO CHART_SONGS VALUES ('CH-003', 1204, 1);
INSERT INTO CHART_SONGS VALUES ('CH-003', 0406, 2);
INSERT INTO CHART_SONGS VALUES ('CH-004', 0309, 1);
INSERT INTO CHART_SONGS VALUES ('CH-004', 2809, 2);
INSERT INTO CHART_SONGS VALUES ('CH-005', 0801, NULL);
INSERT INTO CHART_SONGS VALUES ('CH-006', 2911, 1);
INSERT INTO CHART_SONGS VALUES ('CH-006', 0809, 2);
INSERT INTO CHART_SONGS VALUES ('CH-006', 0910, 3);
INSERT INTO CHART_SONGS VALUES ('CH-007', 3012, 1);
INSERT INTO CHART_SONGS VALUES ('CH-007', 2701, 2);
INSERT INTO CHART_SONGS VALUES ('CH-007', 1005, 3);
INSERT INTO CHART_SONGS VALUES ('CH-008', 1204, NULL);
INSERT INTO CHART_SONGS VALUES ('CH-009', 2005, 1);
INSERT INTO CHART_SONGS VALUES ('CH-009', 2406, 2);
INSERT INTO CHART_SONGS VALUES ('CH-010', 1702, NULL);
INSERT INTO CHART_SONGS VALUES ('CH-011', 1805, 1);
INSERT INTO CHART_SONGS VALUES ('CH-011', 2709, 2);
INSERT INTO CHART_SONGS VALUES ('CH-012', 1007, NULL);
INSERT INTO CHART_SONGS VALUES ('CH-013', 1905, 1);
INSERT INTO CHART_SONGS VALUES ('CH-013', 0309, 2);
INSERT INTO CHART_SONGS VALUES ('CH-013', 1702, 3);
INSERT INTO CHART_SONGS VALUES ('CH-014', 0809, NULL);
INSERT INTO CHART_SONGS VALUES ('CH-015', 2407, 1);
INSERT INTO CHART_SONGS VALUES ('CH-015', 0504, 2);

CREATE TABLE MUSIC_LICENSE (LICENSE_ID NUMBER(4), LICENSE_NAME VARCHAR2(50) NOT NULL, FEE_IN_RS NUMBER(10, 2) NOT NULL CHECK (FEE_IN_RS >= 0), FEE_PERIOD VARCHAR2(50) NOT NULL, LICENSE_AGREEMENT VARCHAR2(1000), CONSTRAINT MUSIC_LICENSE_PK PRIMARY KEY(LICENSE_ID));

INSERT INTO MUSIC_LICENSE VALUES (1, 'Personal Use License', 1500, 'per month', 'Permission granted for personal, non-commercial streaming and enjoyment of the licensed music.');
INSERT INTO MUSIC_LICENSE VALUES (2, 'Commercial Use License', 7500, 'per month', 'Permission granted for the commercial use of the licensed music in establishments or promotional materials.');
INSERT INTO MUSIC_LICENSE VALUES (3, 'Synchronization License', 75000, 'per usage', 'Permission granted for the synchronization of the licensed music with visual content in movies, TV shows, advertisements, or video games.');
INSERT INTO MUSIC_LICENSE VALUES (4, 'Performance License', 15000, 'per event', 'Permission granted for the live performance and public usage of the licensed music.');
INSERT INTO MUSIC_LICENSE VALUES (5, 'Mechanical License', 5000, 'per usage', 'Permission granted for the reproduction and distribution of the licensed music, including cover versions and digital copies.');

INSERT INTO MUSIC_LICENSE VALUES (6, 'Streaming License', 5000, 'per month', 'Permission granted for the streaming and online playback of licensed music on digital platforms and streaming services.');
INSERT INTO MUSIC_LICENSE VALUES (7, 'Digital Distribution License', 1.50, 'per stream/download', 'Permission granted for the digital distribution and streaming of the licensed music through online platforms and services.');
INSERT INTO MUSIC_LICENSE VALUES (8, 'Royalty-Free License', 10000, 'one-time fee', 'Permission granted for the unlimited use of the licensed music without the need for ongoing royalty payments.');
INSERT INTO MUSIC_LICENSE VALUES (9, 'Sample License', 20000, 'per sample usage', 'Permission granted for the use of copyrighted music samples in new compositions or recordings.');
INSERT INTO MUSIC_LICENSE VALUES (10, 'Limited Time/Usage License', 20000, 'for a 30-day period', 'Permission granted for the time-limited or usage-limited access to the licensed music for specific promotional campaigns or limited-time events.');
INSERT INTO MUSIC_LICENSE VALUES (11, 'Broadcast Synchronization License', 75000, 'per usage', 'Permission granted for the synchronization of licensed music with audiovisual content for broadcast purposes, ensuring proper rights and clearances.');
INSERT INTO MUSIC_LICENSE VALUES (12, 'Educational License', 25, 'per student per year', 'Permission granted for the use of the licensed music exclusively for educational purposes within authorized educational institutions, promoting a conducive learning environment.');
INSERT INTO MUSIC_LICENSE VALUES (13, 'On-Hold Music License', 50000, 'per year', 'Permission granted for the use of licensed music as on-hold or background music, enhancing the caller experience while adhering to copyright regulations.');
INSERT INTO MUSIC_LICENSE VALUES (14, 'Karaoke License', 5000, 'per song per year', 'Permission granted for the use of licensed music in karaoke settings, enabling participants to enjoy interactive singing experiences while respecting the rights of artists and creators.');
INSERT INTO MUSIC_LICENSE VALUES (15, 'Public Performance Venue License', 2500, 'per performance', 'Permission granted for the public performance of licensed music at live events, ensuring fair compensation for artists and compliance with legal requirements.');

CREATE TABLE ALBUM_LICENSE (ALBUM_ID VARCHAR2(10) NOT NULL, LICENSE_ID NUMBER(4), START_DATE DATE, END_DATE DATE CHECK (END_DATE >= START_DATE), CONSTRAINT ALBUM_LICENSE_PK PRIMARY KEY (ALBUM_ID, LICENSE_ID), CONSTRAINT ALBUM_LICENSE_ALBUMS_FK FOREIGN KEY(ALBUM_ID) REFERENCES ALBUMS(ALBUM_ID), CONSTRAINT ALBUM_LICENSE_MUSIC_LICENSE_FK FOREIGN KEY(LICENSE_ID) REFERENCES MUSIC_LICENSE(LICENSE_ID));
INSERT INTO ALBUM_LICENSE VALUES ('BTD-2012', 1, '12/12/2012', '12/11/2019');
INSERT INTO ALBUM_LICENSE VALUES ('BTD-2012', 3, '01/01/2019', '04/30/2024');
INSERT INTO ALBUM_LICENSE VALUES ('FLK-2020', 9, '10/08/2021', '10/07/2023');
INSERT INTO ALBUM_LICENSE VALUES ('LYT-2018', 11, '05/18/2022', '08/17/2030');
INSERT INTO ALBUM_LICENSE VALUES ('SA-2022', 5, '06/30/2022', '09/30/2023');
INSERT INTO ALBUM_LICENSE VALUES ('RNS-2022', 8, '07/29/2022', '07/28/2029');
INSERT INTO ALBUM_LICENSE VALUES ('RNS-2022', 5, '07/29/2022', '07/28/2029');
INSERT INTO ALBUM_LICENSE VALUES ('FTF-2022', 8, '02/14/2023', '12/13/2023');
INSERT INTO ALBUM_LICENSE VALUES ('DVD-2017', 12, '11/13/2018', '06/04/2022');
INSERT INTO ALBUM_LICENSE VALUES ('EVR-2020', 5, '08/04/2021', '08/03/2026');
INSERT INTO ALBUM_LICENSE VALUES ('EVR-2020', 3, '12/15/2021', '12/14/2022');
INSERT INTO ALBUM_LICENSE VALUES ('EVR-2020', 6, '02/27/2021', '07/28/2022');
INSERT INTO ALBUM_LICENSE VALUES ('AQ-2015', 4, '03/03/2016', '09/02/2016');
INSERT INTO ALBUM_LICENSE VALUES ('SR-2021', 14, '05/22/2021', '05/21/2029');
INSERT INTO ALBUM_LICENSE VALUES ('SR-2021', 15, '05/22/2021', '05/21/2026');
INSERT INTO ALBUM_LICENSE VALUES ('MOS-2019', 2, '10/05/2020', '10/04/2022');
INSERT INTO ALBUM_LICENSE VALUES ('MOS-2019', 4, '10/19/2020', '10/18/2021');
INSERT INTO ALBUM_LICENSE VALUES ('MOS-2019', 9, '10/24/2020', '04/23/2021');
INSERT INTO ALBUM_LICENSE VALUES ('HRH-2022', 5, '05/23/2022', '05/22/2027');
INSERT INTO ALBUM_LICENSE VALUES ('HRH-2022', 7, '07/02/2022', '07/22/2023');
INSERT INTO ALBUM_LICENSE VALUES ('NIL-2021', 13, '02/23/2022', '08/22/2022');

INSERT INTO ALBUM_LICENSE VALUES ('C&B-2023', 10, '02/17/2023', '02/16/2024');
INSERT INTO ALBUM_LICENSE VALUES ('C&B-2023', 1, '02/18/2023', '02/17/2028');
INSERT INTO ALBUM_LICENSE VALUES ('IWA-2021', 6, '07/31/2022', '01/01/2025');
INSERT INTO ALBUM_LICENSE VALUES ('TMH-2012', 3, '11/12/2012', '11/11/2017');
INSERT INTO ALBUM_LICENSE VALUES ('TMH-2012', 12, '11/23/2012', '11/24/2020');
INSERT INTO ALBUM_LICENSE VALUES ('TMH-2012', 8, '11/29/2012', '11/28/2022');
INSERT INTO ALBUM_LICENSE VALUES ('KK-2020', 3, '04/20/2020', '07/13/2022');

CREATE TABLE FOLLOWERS (ARTIST_ID VARCHAR2(10), FOLLOWER_ID VARCHAR2(20), DATE_FOLLOWED DATE, CONSTRAINT FOLLOWERS_PK PRIMARY KEY (ARTIST_ID, FOLLOWER_ID), CONSTRAINT FOLLOWERS_USERS_FK FOREIGN KEY (FOLLOWER_ID) REFERENCES USERS(USER_ID), CONSTRAINT FOLLOWERS_ARTISTS_FK FOREIGN KEY (ARTIST_ID) REFERENCES ARTISTS(ARTIST_ID));
INSERT INTO FOLLOWERS VALUES ('LDR-1985', 'FNW-100', '01/01/2016');
INSERT INTO FOLLOWERS VALUES ('LDR-1985', 'AD-107', '05/11-2012');
INSERT INTO FOLLOWERS VALUES ('LDR-1985', 'SB-106', '11/22/2014');
INSERT INTO FOLLOWERS VALUES ('LDR-1985', 'AMC-114', '06/30/2023')
INSERT INTO FOLLOWERS VALUES ('OR-2003', 'CH-109', '12/21/2021');
INSERT INTO FOLLOWERS VALUES ('TS-1989', 'AK-103', '03/02/2017');
INSERT INTO FOLLOWERS VALUES ('TS-1989', 'CG-118', '01/19/2022');
INSERT INTO FOLLOWERS VALUES ('TV-2012', 'AK-103', '06/10/2021');
INSERT INTO FOLLOWERS VALUES ('BTS-2010', 'MA-105', '03/03/2021');
INSERT INTO FOLLOWERS VALUES ('BTS-2010', 'AS-104', '03/03/2021');
INSERT INTO FOLLOWERS VALUES ('AQ-1988', 'HH-110', '11/11/2011');
INSERT INTO FOLLOWERS VALUES ('ZM-1993', 'TQZ-101', '04/21/2022');
INSERT INTO FOLLOWERS VALUES ('ZM-1993', 'JJ-115', '06/12/2020');
INSERT INTO FOLLOWERS VALUES ('MA-2016', 'OZ-113', '06/17/2019');
INSERT INTO FOLLOWERS VALUES ('LT-1991', 'KK-108', '09/08/2017');
INSERT INTO FOLLOWERS VALUES ('LT-1991', 'AK-103', '03/30/2020');
INSERT INTO FOLLOWERS VALUES ('B-1981', 'SB-106', '07/12/2018');
INSERT INTO FOLLOWERS VALUES ('B-1981', 'FNW-100', '02/02/2021');
INSERT INTO FOLLOWERS VALUES ('B-1981', 'WL-116', '11/05/2016');
INSERT INTO FOLLOWERS VALUES ('1D-2010', 'AK-103', '04/19/2017');
INSERT INTO FOLLOWERS VALUES ('1D-2010', 'CG-118', '08/28/2019');
INSERT INTO FOLLOWERS VALUES ('IN-2013', 'AK-103', '06/13/2022');
INSERT INTO FOLLOWERS VALUES ('IN-2013', 'EH-119', '01/02/2018');
INSERT INTO FOLLOWERS VALUES ('HS-1994', 'PR-117', '05/28/2019');
INSERT INTO FOLLOWERS VALUES ('HS-1994', 'RJL-102', '10/17/2021');
INSERT INTO FOLLOWERS VALUES ('ES-1991', 'WL-116', '07/29/2020');
INSERT INTO FOLLOWERS VALUES ('CG-1998', 'AK-103', '11/10/2017');
INSERT INTO FOLLOWERS VALUES ('MA-2016', 'AK-103', '03/14/2018');
INSERT INTO FOLLOWERS VALUES ('NA-1991', 'IR-112', '09/03/2022');
INSERT INTO FOLLOWERS VALUES ('JK-1997', 'MA-105', '06/09/2021');