MILESTONE-1 DATABASE DESIGN FOR MUSIC STREAMING APP

NAME: VEER KUMAR

I. INSERT STATEMENTS

##1 INSERT STATEMENT FOR ARTISTS:

```
INSERT INTO project.artists (artist_id, artist_name) VALUES (2, 'Michael Jackson');
INSERT INTO project.artists (artist_id, artist_name) VALUES (3, 'Queen');
INSERT INTO project.artists (artist_id, artist_name) VALUES (4, 'Led Zeppelin');
INSERT INTO project.artists (artist_id, artist_name) VALUES (5, 'Pink Floyd');
INSERT INTO project.artists (artist_id, artist_name) VALUES (6, 'Elvis Presley');
INSERT INTO project.artists (artist_id, artist_name) VALUES (7, 'The Rolling Stones');
INSERT INTO project.artists (artist_id, artist_name) VALUES (8, 'Bob Dylan');
INSERT INTO project.artists (artist_id, artist_name) VALUES (9, 'David Bowie');
INSERT INTO project.artists (artist_id, artist_name) VALUES (10, 'Prince');
INSERT INTO project.artists (artist_id, artist_name) VALUES (11, 'Nirvana');
INSERT INTO project.artists (artist_id, artist_name) VALUES (12, 'Radiohead');
INSERT INTO project.artists (artist_id, artist_name) VALUES (13, 'U2');
INSERT INTO project.artists (artist_id, artist_name) VALUES (14, 'Metallica');
INSERT INTO project.artists (artist_id, artist_name) VALUES (15, 'AC/DC');
```

##2 INSERTING INTO GENRES TABLE:

```
INSERT INTO project.genres (genre_name) VALUES ('Pop');
INSERT INTO project.genres (genre_name) VALUES ('Rock');
INSERT INTO project.genres (genre_name) VALUES ('R&B');
INSERT INTO project.genres (genre_name) VALUES ('Hip-hop');
INSERT INTO project.genres (genre_name) VALUES ('Electronic');
INSERT INTO project.genres (genre_name) VALUES ('Funk');
INSERT INTO project.genres (genre_name) VALUES ('Soul');
```

```
INSERT INTO project.genres (genre_name) VALUES ('Disco');

INSERT INTO project.genres (genre_name) VALUES ('Blues');

INSERT INTO project.genres (genre_name) VALUES ('Country');

INSERT INTO project.genres (genre_name) VALUES ('Reggae');

INSERT INTO project.genres (genre_name) VALUES ('Jazz');

INSERT INTO project.genres (genre_name) VALUES ('Classical');
```

#had to populate these two tables before albums because it has two foreign keys referencing genres and artists

<u>##3 INSERT INTO ALBUMS:</u>

```
INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (1, 'Please Please Me', 1, 1963, 'Pop');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (2, 'With the Beatles', 1, 1963, 'Rock');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (3, 'A Hard Days Night', 1, 1964, 'Rock');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (4, 'Beatles for Sale', 1, 1964, 'Pop');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (5, 'Help!', 1, 1965, 'Pop');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)

VALUES (6, 'Rubber Soul', 1, 1965, 'Rock');


INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
```

```
VALUES (7, 'Revolver', 1, 1966, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (8, 'Sgt. Peppers Lonely Hearts Club Band', 1, 1967, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (9, 'Magical Mystery Tour', 1, 1967, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (10, 'The Beatles (White Album)', 1, 1968, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (11, 'Yellow Submarine', 1, 1969, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (12, 'Abbey Road', 1, 1969, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (13, 'Let It Be', 1, 1970, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (14, 'Live at the BBC', 1, 1994, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (15, 'Anthology 1', 1, 1995, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (16, 'Anthology 2', 1, 1996, 'Rock');
```

```
INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (17, 'Anthology 3', 1, 1996, 'Rock');

INSERT INTO project.albums (album_id, album_name, artist_id, year, genre_name)
VALUES (18, 'Let It Be...Naked', 1, 2003, 'Rock');
```

##4 NOW WE INSERT INTO TRACKS SINCE IT HAS FOREIGN KEY REFERENCING ALBUMS

```
INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (2, 'Drive My Car', 1, 2, 02.25, 4.7);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (3, 'Norwegian Wood', 1, 3, 02.01, 4.6);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (4, 'Michelle', 1, 4, 02.44, 4.9);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (5, 'In My Life', 1, 5, 02.26, 4.8);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (6, 'The Way You Make Me Feel', 2, 1, 04.58, 4.7);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (7, 'Man in the Mirror', 2, 2, 05.19, 4.9);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (8, 'Dirty Diana', 2, 3, 04.41, 4.8);
```

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (9, 'Smooth Criminal', 2, 4, 04.18, 4.9);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (10, 'Black or White', 2, 5, 04.15, 4.8);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (11, 'Billie Jean', 2, 6, 04.54, 4.9);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (12, 'Stairway to Heaven', 3, 1, 08.02, 4.9);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (13, 'Black Dog', 3, 2, 04.54, 4.8);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (14, 'Rock and Roll', 3, 3, 03.41, 4.7);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (15, 'Going to California', 3, 4, 03.31, 4.6);

INSERT INTO project.tracks (track_id, track_name, album_id, track_number, duration, rating)
VALUES (16, 'Kashmir', 3, 5, 08.37, 4.9);

<u>##5 INSERT INTO TABLE USERS:</u>

INSERT INTO project.users (user_id, name, email, date_of_birth, is_premium, account_created_on, password) VALUES

(1, 'John Smith', 'johnsmith@yahoo.com', '1990-05-15', 'yes', '2022-01-01', 'mypassword1'),

```
    (2, 'Emily Brown', 'emilybrown@gmail.com', '1995-09-20', 'no', '2023-02-05', 'mypassword2'),
    (3, 'David Lee', 'davidlee@yahoo.com', '1985-12-10', 'yes', '2023-03-10', 'mypassword3'),
    (4, 'Sophia Davis', 'sophiadavis@reddit.com', '2000-03-25', 'no', '2022-04-15', 'mypassword4'),
    (5, 'Max Wilson', 'maxwilson@gmail.com', '1992-07-18', 'yes', '2021-05-20', 'mypassword5');
```

##6 INSERT INTO PLAYLISTS:

```
INSERT INTO project.playlists (playlist_id, playlist_name, user_id) VALUES
    (1, 'Rock Classics',1),
    (2, '80s Hits', 2),
    (3, 'Pop Anthems',3),
    (4, 'Country Music',4),
    (5, 'Hip Hop Mix',5);
```

##7 INSERT INTO PLAYLIST_TRACKS/BRIDGE TABLE:

```
INSERT INTO project.playlist_tracks (playlist_id, track_id) VALUES
    (1, 1),
    (1, 2),
    (2, 3),
    (2, 4),
    (3, 5);
```

##8 INSERT INTO STREAMING_TIME TABLE:

```
INSERT INTO project.streaming_time (user_id, track_id, streaming_time)
VALUES (1, 1, tstzrange('2023-03-22 12:00:00', '2023-03-22 12:30:00', '[)')),
        (2, 3, tstzrange('2023-03-21 20:00:00', '2023-03-21 20:15:00', '[)')),
        (1, 4, tstzrange('2023-03-22 08:00:00', '2023-03-22 08:45:00', '[)')),
        (3, 2, tstzrange('2023-03-20 15:30:00', '2023-03-20 16:00:00', '[)')),
        (4, 5, tstzrange('2023-03-23 10:15:00', '2023-03-23 11:00:00', '[)'));
```

II. UPDATE STATEMENTS:

1) Let's say a user wants to update the name of one of their playlists, they could use the following update statement:

```
UPDATE project.playlists
SET playlist_name = 'New Playlist Name'
WHERE playlist_id = 1 AND user_id = 1;
```

2) Update the email address of a user with user_id 1:

```
UPDATE project.users
SET email = 'newemail@example.com'

WHERE user_id = 1;
```

3) Update the rating of a track with track_id 5:

```
UPDATE project.tracks
SET rating = 4.5

WHERE track_id = 5;
```

4) Update the start and end times of a streaming session for user_id 3 and track_id 2:

```
UPDATE project.streaming_time
SET streaming_time = tstzrange('2023-03-20 15:30:00', '2023-03-20 16:15:00', '[)')

WHERE user_id = 3 AND track_id = 2;
```

III. DELETE STATEMENTS:

1) Delete a specific track from the tracks table:

```
DELETE FROM project.tracks
WHERE track_id = 1;
```

2) Delete all tracks that belong to an album:

```
DELETE FROM project.tracks
WHERE album_id = 5;
```

3) Delete a user from the users table:

```
DELETE FROM project.users

WHERE user_id = 3;
```

4) Delete a playlist from the playlists table:

```
DELETE FROM project.playlists

WHERE playlist_id = 4;
```

5) Delete a playlist track from the playlist_tracks table:

```
DELETE FROM project.playlist_tracks
WHERE playlist_id = 1 AND track_id = 2;
```

IV. SELECT STATEMENTS:

1) Retrieve all albums with their corresponding artist and genre names:

```
WITH album_info AS (
   SELECT albums.album_name, artists.artist_name, genres.genre_name
   FROM project.albums
   INNER JOIN project.artists ON albums.artist_id = artists.artist_id
   INNER JOIN project.genres ON albums.genre_name = genres.genre_name
)
SELECT * FROM album_info;
```

Output:

Data Output Messages Notifications

	album_name character varying (50)	artist_name character varying (35)	genre_name character varying (35)
1	Please Please Me	The Beatles	Pop
2	With the Beatles	The Beatles	Rock
3	A Hard Days Night	The Beatles	Rock
4	Beatles for Sale	The Beatles	Pop
5	Help!	The Beatles	Pop
6	Rubber Soul	The Beatles	Rock
7	Revolver	The Beatles	Rock

Total rows: 18 of 18 Query complete 00:00:00.104

2) Retrieve the average duration of tracks for each album:

```
WITH avg_durations AS (
   SELECT albums.album_name, AVG(tracks.duration) AS avg_duration
   FROM project.tracks
   INNER JOIN project.albums ON tracks.album_id = albums.album_id
   GROUP BY albums.album_name
```

```
)
SELECT * FROM avg_durations;
```

Output:

Data Output Messages Notifications

	album_name character varying (50)	avg_duration numeric
1	Please Please Me	2.2000000000000000
2	A Hard Days Night	5.5300000000000000
3	With the Beatles	4.5083333333333333

3) Retrieving the top 5most streamed tracks along with their corresponding album names:

```
WITH top_tracks AS (
   SELECT tracks.track_name, albums.album_name, COUNT(*) AS stream_count
   FROM project.streaming_time
   INNER JOIN project.tracks ON streaming_time.track_id = tracks.track_id
   INNER JOIN project.albums ON tracks.album_id = albums.album_id
   GROUP BY tracks.track_name, albums.album_name
   ORDER BY stream_count DESC
   LIMIT 5
)
SELECT * FROM top_tracks;
```

Output:

	track_name character varying (35)	album_name character varying (50)	stream_count bigint
1	Drive My Car	Please Please Me	1
2	In My Life	Please Please Me	1
3	Yesterday	Please Please Me	1
4	Norwegian Wood	Please Please Me	1
5	Michelle	Please Please Me	1

4) Using Cross-join to return a result set that combines every user's name with every playlist name:

```
SELECT u.name, p.playlist_name
FROM project.users u
CROSS JOIN project.playlists p;
```

Output:

	name character varying (25)	playlist_name character varying (35)
1	John Smith	Rock Classics
2	Emily Brown	Rock Classics
3	David Lee	Rock Classics
4	Sophia Davis	Rock Classics
5	Max Wilson	Rock Classics
6	John Smith	80s Hits
7	Emily Brown	80s Hits

Total rows: 25 of 25 Query complete 00:00:00.095

5) An example SELECT statement using generate_series() on the project.albums table:

```
SELECT *
FROM project.albums
WHERE year IN (SELECT generate_series(1960, 1964));
```

Output:

Data Output Messages Notifications

	album_id [PK] smallint	album_name character varying (50)	artist_id smallint	year smallint	genre_name character varying (50)
1	1	Please Please Me	1	1963	Pop
2	2	With the Beatles	1	1963	Rock
3	3	A Hard Days Night	1	1964	Rock
4	4	Beatles for Sale	1	1964	Pop

➢ Potential Contigencies one may encounter:

1) This database has multiple foreign key restrictions, therefore removing data from some tables may lead to issues if the removed data is used as a reference by other tables.
For instance, because the user id is used as a reference by the playlists table, deleting a user from the users database who has produced playlists that are still in the playlists table would result in a foreign key constraint problem.

```
DELETE FROM project.users WHERE user_id = 1;
```

Output:

Data Output Messages Notifications

```
ERROR:  update or delete on table "users" violates foreign key constraint "fk_playlists_users" on table "playlists"
DETAIL:  Key (user_id)=(1) is still referenced from table "playlists".
SQL state: 23503
```

2) Trying to delete a genre from the genres table that is still referenced by albums in the albums table. Here is an example of a delete statement that would result in a foreign key constraint error:
DELETE FROM project.genres WHERE genre_name = 'Rock';

Output:

Data Output Messages Notifications

```
ERROR:  update or delete on table "genres" violates foreign key constraint "fk_albums_genre_name" on table "albums"
DETAIL:  Key (genre_name)=(Rock) is still referenced from table "albums".
SQL state: 23503
```