SQL Task 6

First of all, we need to create a database name as soundcloud_db

```
Create database soundcloud_db ;
Use soundcloud_db ;
```

Now we will create table name as users.

```
CREATE TABLE users (
    user_id INTEGER PRIMARY KEY,
    username VARCHAR(100),
    email VARCHAR(200),
    join_date DATE,
    country VARCHAR(200)
);
```

We will insert some values in our table.

user_id	username	email	join_date	country
1	user1	user1@example.com	2023-08-02	Japan
2	user2	user2@example.com	2020-08-16	Brazil
3	user3	user3@example.com	2020-02-21	France
4	user4	user4@example.com	2024-02-27	Japan
5	user5	user5@example.com	2021-07-17	UK
6	user6	user6@example.com	2021-05-16	France
7	user7	user7@example.com	2021-04-02	France
8	user8	user8@example.com	2020-10-12	Japan
9	user9	user9@example.com	2024-02-17	Germany
10	user10	user10@example.com	2020-07-28	UK

Now we will create another table name as playlists.

```
CREATE TABLE playlists (
    playlist_id INTEGER PRIMARY KEY,
    user_id INTEGER,
    name VARCHAR(200),
    created_at DATE,
    num_tracks INT,
    FOREIGN KEY (user_id)
        REFERENCES users (user_id)
);
```

Now we will insert some values in our table.

playlist_id	user_id	name	created_at	num_tracks
201	85	Playlist 201	2021-06-08	16
202	15	Playlist 202	2022-08-09	9
203	65	Playlist 203	2022-04-26	16
204	76	Playlist 204	2022-07-11	15
205	71	Playlist 205	2022-11-19	9
206	15	Playlist 206	2022-10-17	12
207	21	Playlist 207	2024-01-12	10
208	19	Playlist 208	2021-06-15	12
209	60	Playlist 209	2021-10-11	6
210	93	Playlist 210	2021-11-05	7

Now we will create another table name as tracks.

```
CREATE TABLE tracks (
    track_id INT PRIMARY KEY,
    user_id INT,
    title VARCHAR(100),
    genre VARCHAR(50),
    upload_date DATE,
    plays INT,
    duration_sec INT,
    FOREIGN KEY (user_id)
        REFERENCES users (user_id)
);
```

Now we will insert some values in above table.

track_id	user_id	title	genre	upload_date	plays	duration_sec
101	72	Track 101	Synthwave	2021-05-27	10689	268
102	78	Track 102	Hip-Hop	2024-06-01	14018	200
103	87	Track 103	Synthwave	2024-07-24	20219	334
104	62	Track 104	Trap	2021-12-17	27546	317
105	40	Track 105	House	2023-12-30	13146	319
106	85	Track 106	Hip-Hop	2022-05-17	24918	144
107	80	Track 107	Rock	2021-12-01	35458	208
108	82	Track 108	Rock	2023-08-05	4424	387
109	53	Track 109	Pop	2023-02-17	10887	262
110	24	Track 110	Techno	2022-07-07	9211	178

Now we will use subqueries in our tables.

List tracks that belong to playlists created by users from the USA.

```
SELECT
    title
FROM
    tracks
WHERE
    track_id IN (SELECT
            track_id
        FROM
            tracks
                JOIN
            users ON users.user_id = tracks.user_id
        WHERE
            country = 'USA');
```

Find the most popular track (highest plays) in each playlist.

```
SELECT
    playlists.playlist_id, playlists.user_id, tracks.title
FROM
    playlists
        JOIN
    users ON users.user_id = playlists.user_id
        JOIN
    tracks ON playlists.user_id = tracks.user_id
WHERE
    tracks.plays = (SELECT
            MAX(plays)
        FROM
            tracks);
```

List tracks longer than the average track duration

```
SELECT
    tracks.title, tracks.duration_sec
FROM
    tracks
WHERE
    tracks.duration_sec > (SELECT
            AVG(tracks.duration_sec)
        FROM
            tracks);
```

Find the most popular track (highest plays) in each playlist.

```
SELECT
    playlists.playlist_id, playlists.name
FROM
    playlists
WHERE
    (SELECT
            COUNT(num_tracks)
        FROM
            playlists);
```

EER diagram.