**Database Design Project**

**Database Modeling for a Music Streaming Service with Artist, Listener, Album, and Single entities.**

**By Georges Machhour and Michel Abou-Assaly**

**Descriptive text:**

In an Entity-Relationship (ER) diagram for a music streaming service like Spotify, Anghami, and Apple Music, there exists a scenario where the primary entities are "Artist" and "Listener", with supporting entities of "Album" and "Single". The "Artist" entity is characterized by attributes such as "Artist ID", "Artist name", "About paragraph", "Discography", "Popular releases", and "Number of monthly listeners". The "Listener" entity is associated with attributes such as "Listener ID", "Listener username", "Liked songs", and "Playlists".

The "Single" entity comprises of attributes such as "Single ID", "Single Name", "Single duration", and "genre".

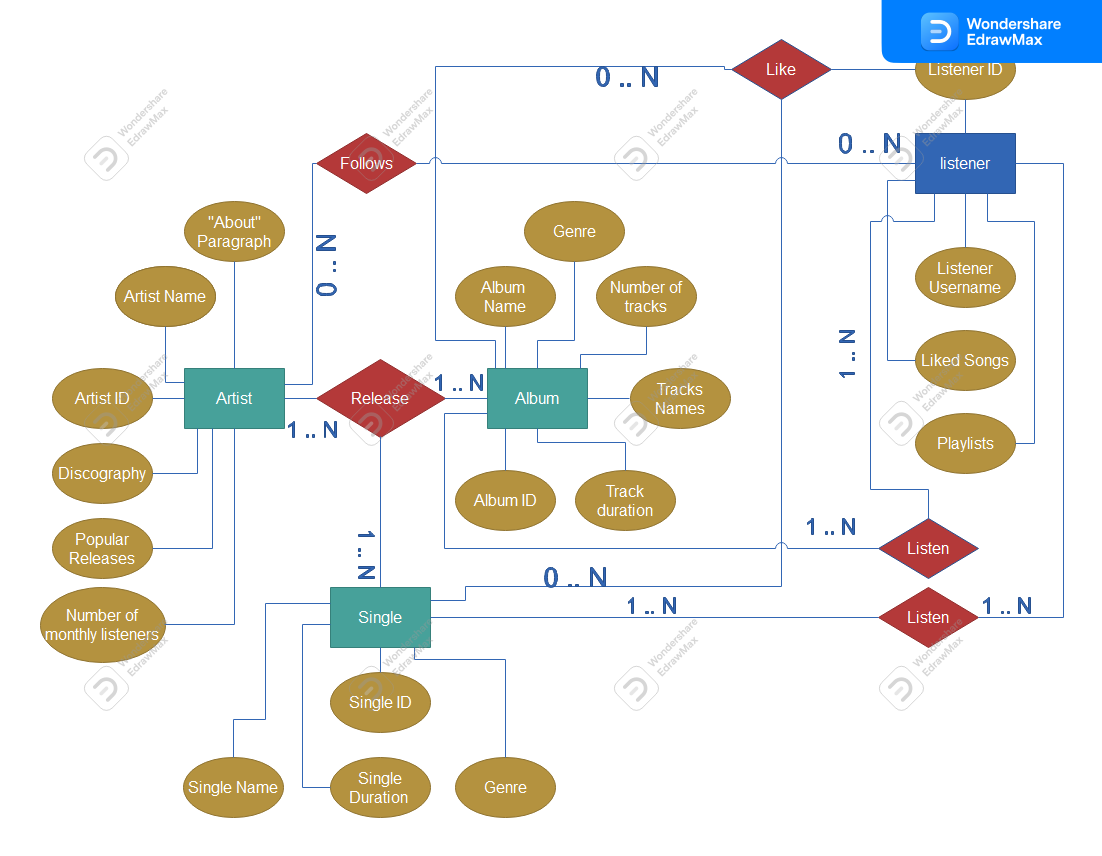
Where the "Album" entity contains multiple tracks and is identified by attributes such as "Album ID", "Album name", "Number of tracks", "Tracks names", "Track duration", and "Genre".

The "Artist" can release an "Album" that contains several tracks or a "Single" that comprises of a single track.

The "Listener" has the ability to stream to either one or multiple "Single(s)" and also to one or multiple tracks within an "Album".

The "Listener" can choose to follow one or more artists or none, and can stream a "Single" track or multiple tracks from an "Album".

The "Listener" can also like one or more "Single(s)", one or more "Album(s)", or/and one or more tracks in that "Album" and these create a "Liked Songs" playlist. Additionally, the "Listener" can create custom "Playlists" where he can add tracks to them.

**ER Diagram: **

**Mapping:**

**Artist** (ArtistsID , Artist Name, About Paragraph, Discography, Popular releases, Number of monthly listeners )

**Listener** (ListenerID , Listener username, Liked songs\* , playlists\*)

**Single** (SingleID , Single Name, Single duration, genre, Artist ID\*)

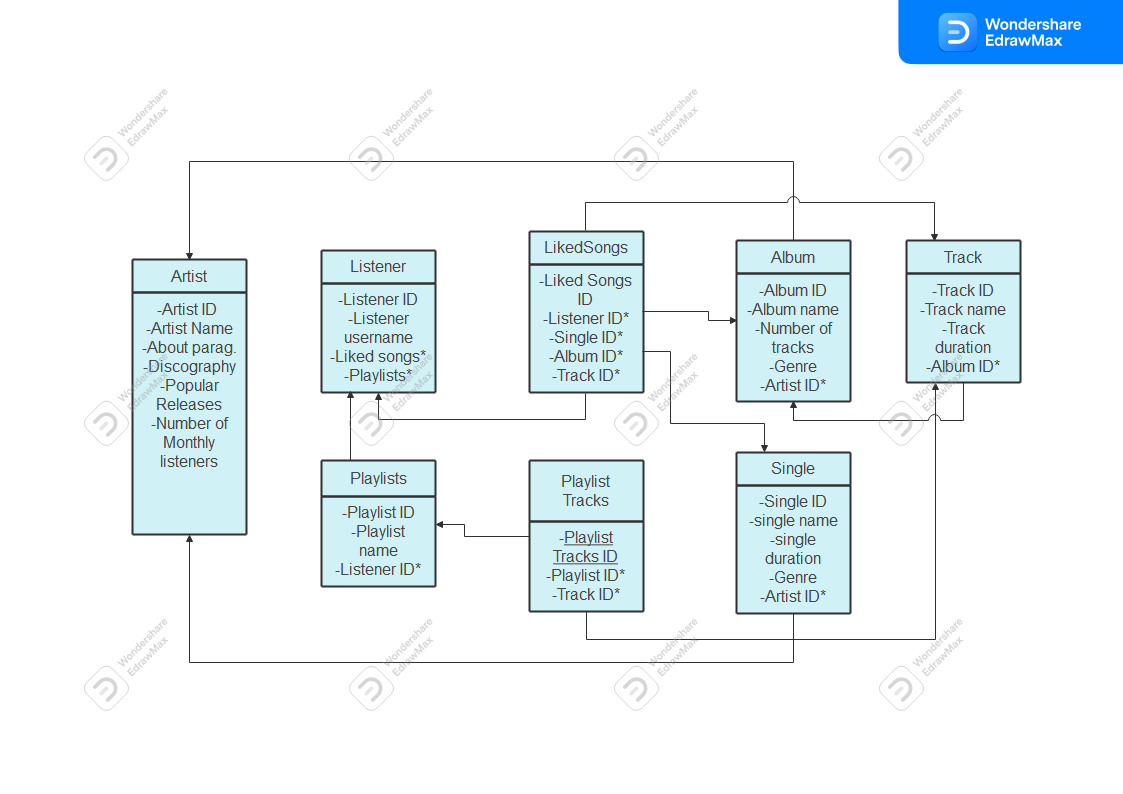
**Album** (AlbumID, Album name, Number of Tracks, Genre, Artist ID\*)

**Track** (TrackID , Track name, Track duration, Album ID\*)

**LikedSongs** (LikedSongsID , Listener ID\*, Single ID\*, Album ID\*, Track ID\*)

**Playlists** (PlaylistID , Playlist name, Listener ID\*)

**PlaylistTracks** (PlaylistTracksID , Playlist ID\*, Track ID\*)

**Relational Model:**

Creating tables in SQL:

CREATE TABLE Artist (  
  ArtistID INT PRIMARY KEY,  
  ArtistName VARCHAR(255),  
  AboutParagraph TEXT,  
  PopularReleases TEXT,  
  MonthlyListeners INT  
);

CREATE TABLE Listener (  
  ListenerID INT PRIMARY KEY,  
  Username VARCHAR(255)  
  
);

CREATE TABLE Single (  
  SingleID INT PRIMARY KEY,  
  SingleName VARCHAR(255),  
  SingleDuration INT,  
  Genre VARCHAR(255),  
  ArtistID INT,  
  FOREIGN KEY (ArtistID) REFERENCES Artist(ArtistID)  
);

CREATE TABLE Album (  
  AlbumID INT PRIMARY KEY,  
  AlbumName VARCHAR(255),  
  NumTracks INT,  
  Genre VARCHAR(255),  
  ArtistID INT,  
  FOREIGN KEY (ArtistID) REFERENCES Artist(ArtistID)  
);

CREATE TABLE Track (  
  TrackID INT PRIMARY KEY,  
  TrackName VARCHAR(255),  
  TrackDuration INT,  
  AlbumID INT,  
  FOREIGN KEY (AlbumID) REFERENCES Album(AlbumID)  
);

CREATE TABLE LikedSongs (  
  LikedSongsID INT PRIMARY KEY,  
  ListenerID INT,  
  SingleID INT,  
  AlbumID INT,  
  TrackID INT,  
  FOREIGN KEY (ListenerID) REFERENCES Listener(ListenerID),  
  FOREIGN KEY (SingleID) REFERENCES Single(SingleID),  
  FOREIGN KEY (AlbumID) REFERENCES Album(AlbumID),  
  FOREIGN KEY (TrackID) REFERENCES Track(TrackID)  
);

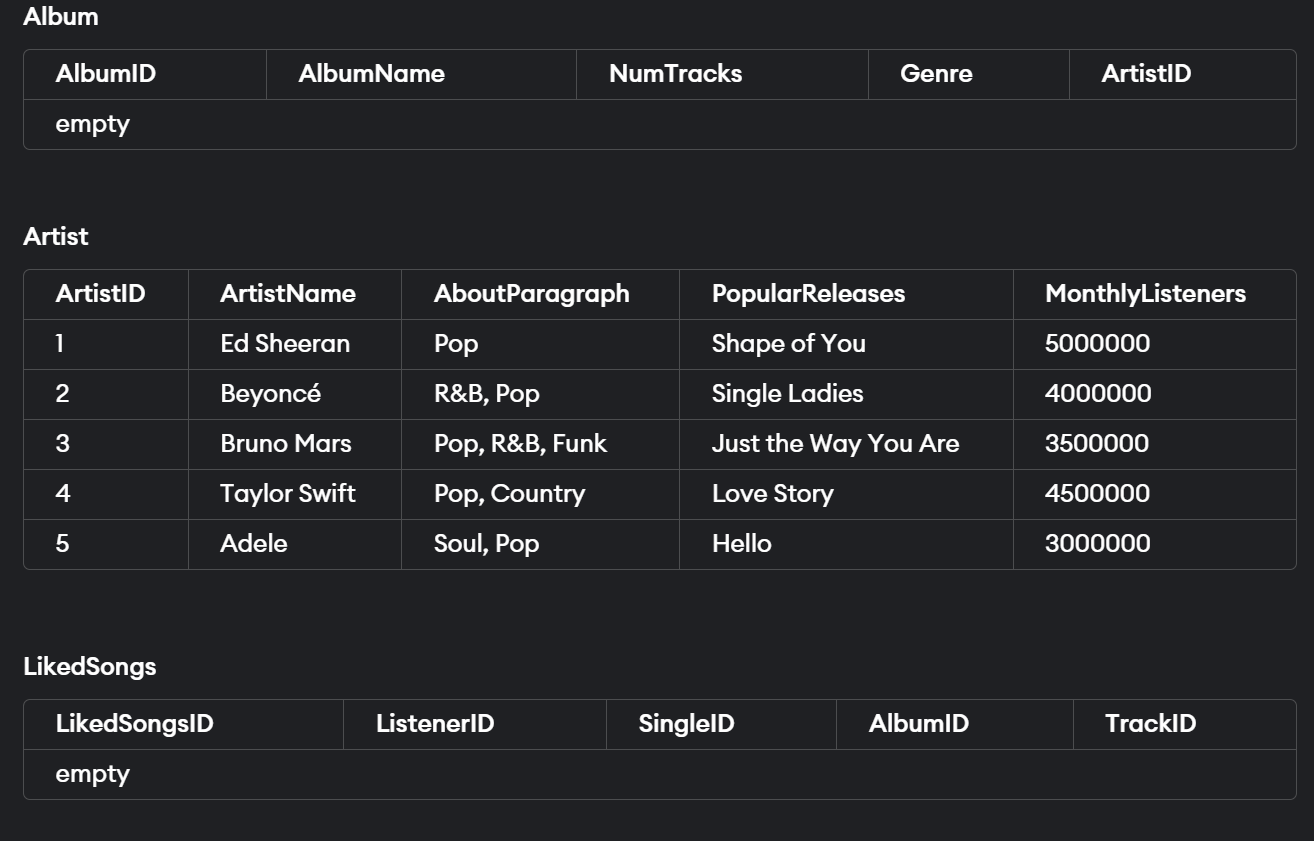
CREATE TABLE Playlists (  
  PlaylistID INT PRIMARY KEY,  
  PlaylistName VARCHAR(255),  
  ListenerID INT,  
  FOREIGN KEY (ListenerID) REFERENCES Listener(ListenerID)  
);

CREATE TABLE PlaylistTracks (  
  PlaylistTracksID INT PRIMARY KEY,  
  PlaylistID INT,  
  TrackID INT,  
  FOREIGN KEY (PlaylistID) REFERENCES Playlists(PlaylistID),  
  FOREIGN KEY (TrackID) REFERENCES Track(TrackID)  
);

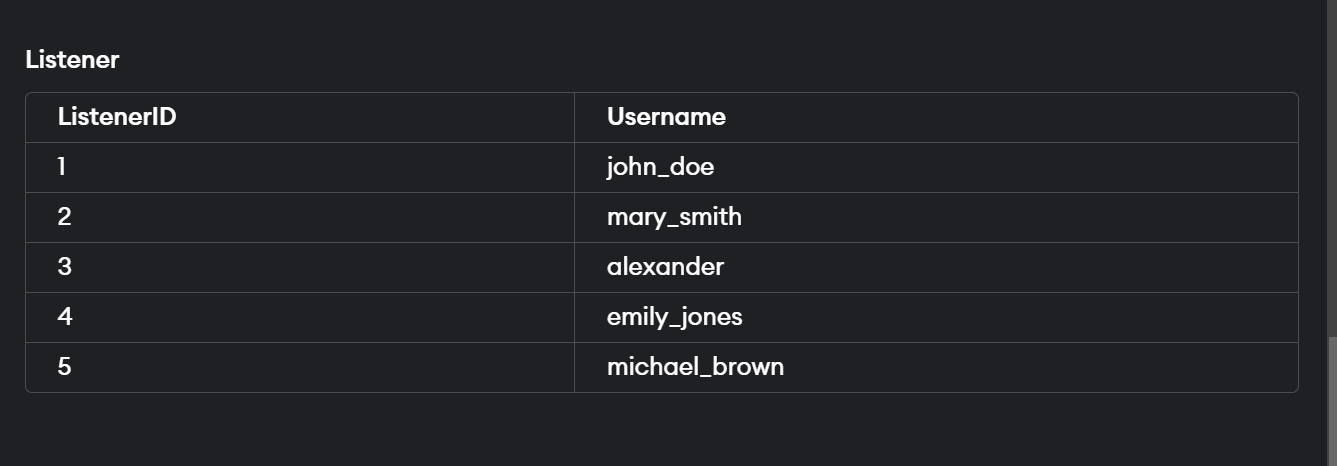
Tables are now created:



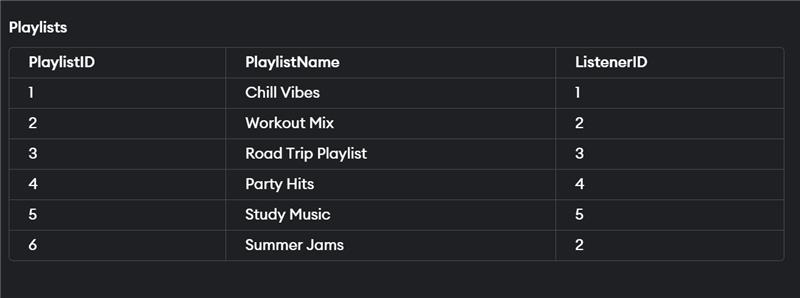
**INSERT INTO Artist (ArtistID, ArtistName, AboutParagraph, PopularReleases, MonthlyListeners)  
VALUES  
  (1, 'Ed Sheeran', 'Pop', 'Shape of You', 5000000),  
  (2, 'Beyoncé', 'R&B, Pop', 'Single Ladies', 4000000),  
  (3, 'Bruno Mars', 'Pop, R&B, Funk', 'Just the Way You Are', 3500000),  
  (4, 'Taylor Swift', 'Pop, Country', 'Love Story', 4500000),  
  (5, 'Adele', 'Soul, Pop', 'Hello', 3000000);  
(6, 'Ed Sheeran', 'Pop', 'Thinking Out Loud', 5000000),  
  (7, 'Ed Sheeran', 'Pop', 'Perfect', 5000000),**



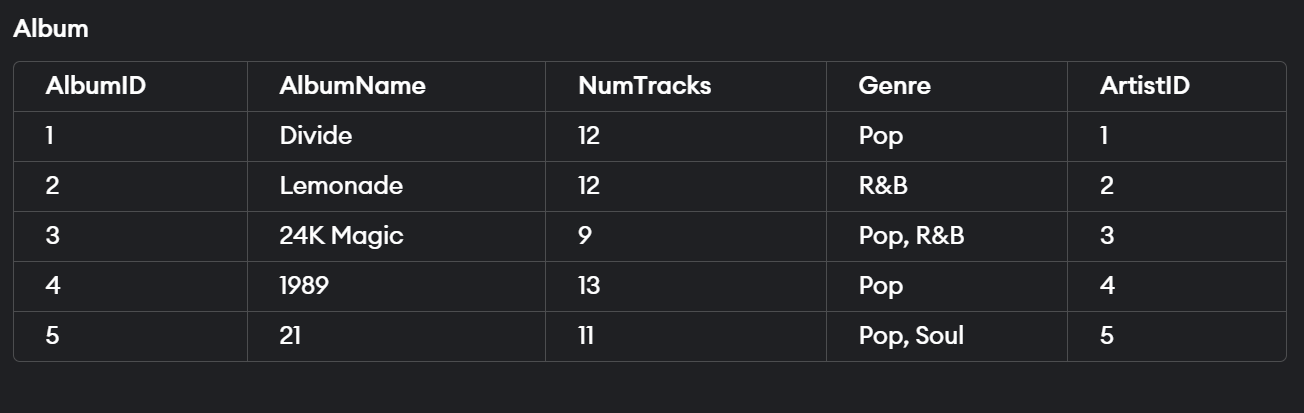
**INSERT INTO Listener (ListenerID, Username)  
VALUES  
  (1, 'john\_doe'),  
  (2, 'mary\_smith'),  
  (3, 'alexander'),  
  (4, 'emily\_jones'),  
  (5, 'michael\_brown');**



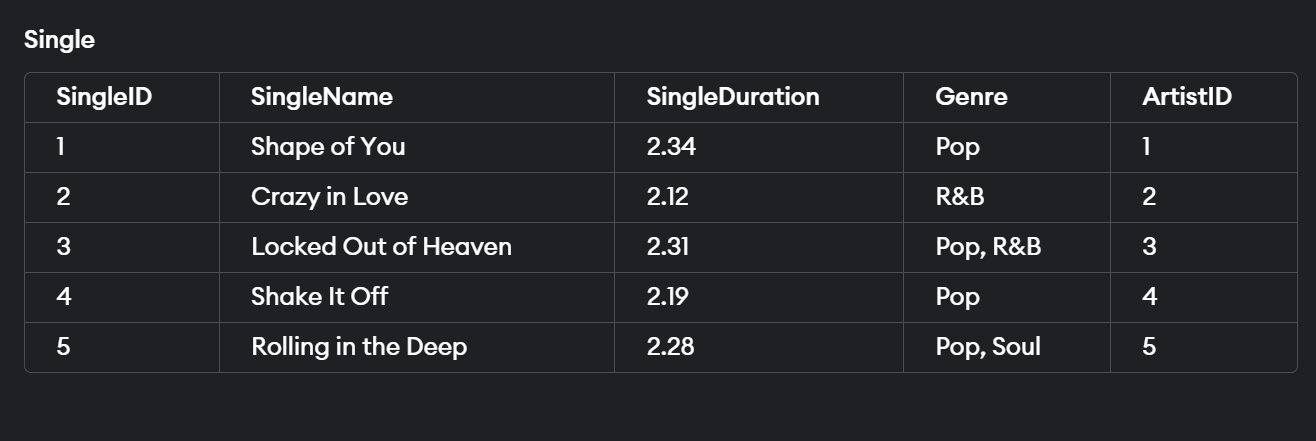
**INSERT INTO Playlists (PlaylistID, PlaylistName, ListenerID)  
VALUES  
  (1, 'Chill Vibes', 1),  
  (2, 'Workout Mix', 2),  
  (3, 'Road Trip Playlist', 3),  
  (4, 'Party Hits', 4),  
  (5, 'Study Music', 5),  
  (6, 'Summer Jams', 2);**



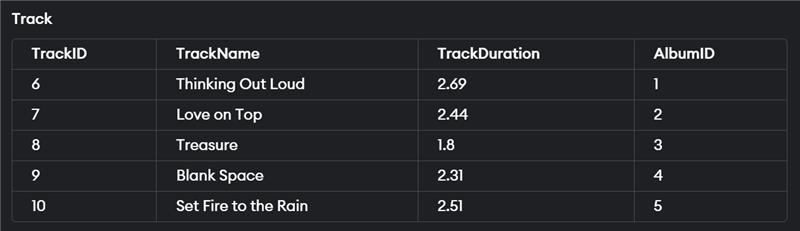
**INSERT INTO Album (AlbumID, AlbumName, NumTracks, Genre, ArtistID)  
VALUES  
  (1, 'Divide', 12, 'Pop', 1),  
  (2, 'Lemonade', 12, 'R&B', 2),  
  (3, '24K Magic', 9, 'Pop, R&B', 3),  
  (4, '1989', 13, 'Pop', 4),  
  (5, '21', 11, 'Pop, Soul', 5);**



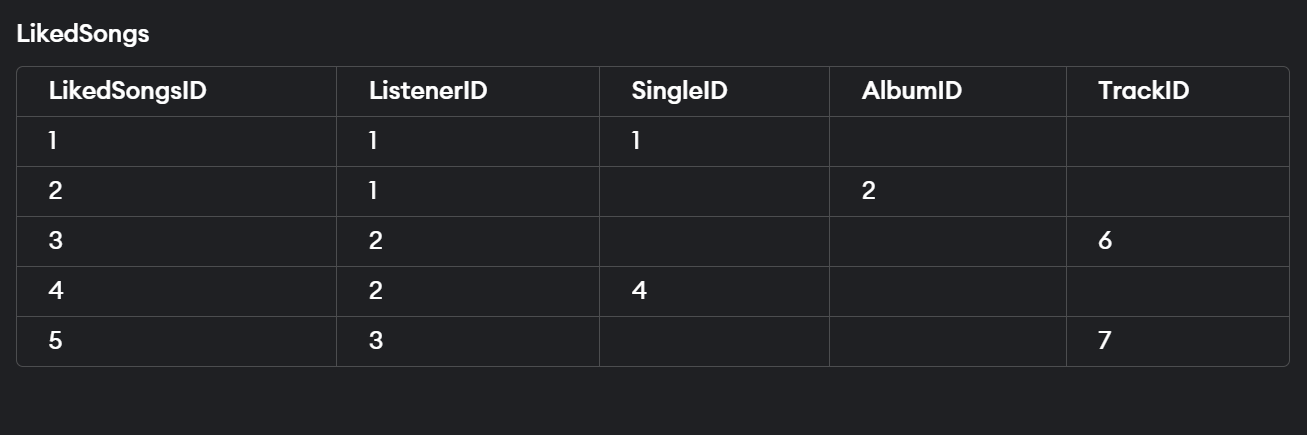
**INSERT INTO Single (SingleID, SingleName, SingleDuration, Genre, ArtistID)  
VALUES  
  (1, 'Shape of You', 2.34, 'Pop', 1),  
  (2, 'Crazy in Love', 2.12, 'R&B', 2),  
  (3, 'Locked Out of Heaven', 2.31, 'Pop, R&B', 3),  
  (4, 'Shake It Off', 2.19, 'Pop', 4),  
  (5, 'Rolling in the Deep', 2.28, 'Pop, Soul', 5);**



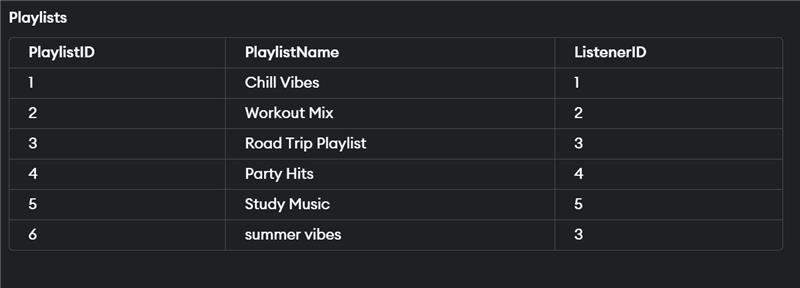
**INSERT INTO Track (TrackID, TrackName, TrackDuration, AlbumID)  
VALUES  
  (6, 'Thinking Out Loud', 2.69, 1),  
  (7, 'Love on Top', 2.44, 2),  
  (8, 'Treasure', 1.80, 3),  
  (9, 'Blank Space', 2.31, 4),  
  (10, 'Set Fire to the Rain', 2.51, 5);**



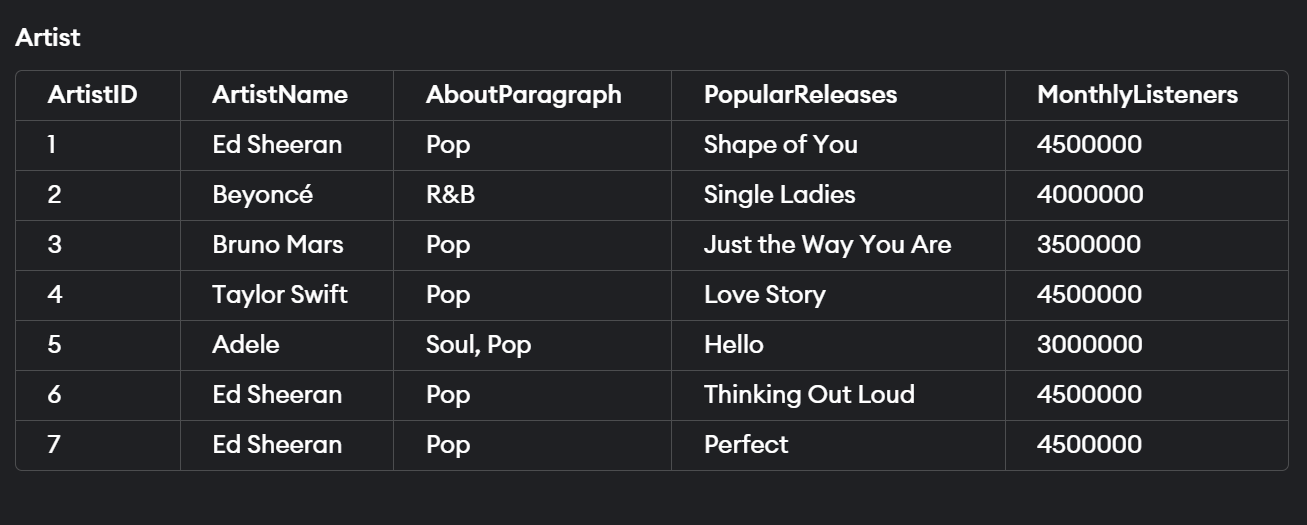
**INSERT INTO LikedSongs (LikedSongsID, ListenerID, SingleID, AlbumID, TrackID)  
VALUES  
  (1, 1, 1, NULL, NULL),  
  (2, 1, NULL, 2, NULL),  
  (3, 2, NULL, NULL, 6),  
  (4, 2, 4, NULL, NULL),  
  (5, 3, NULL, NULL, 7);**



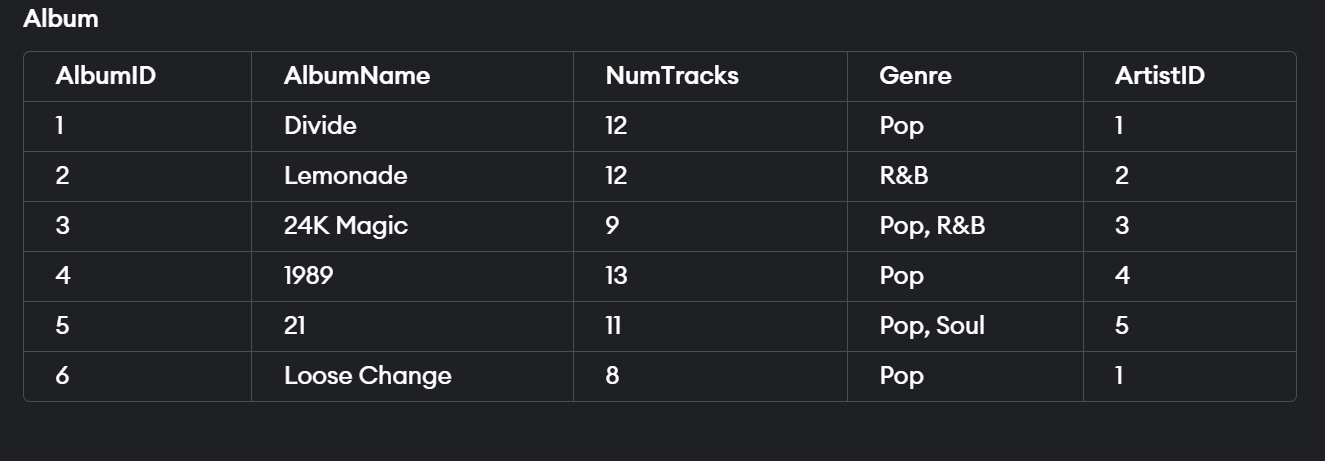
**UPDATE Playlists  
SET PlaylistName = 'summer vibes', ListenerID = 3  
WHERE PlaylistID = 6;**



**UPDATE Artist  
SET MonthlyListeners= 4500000  
WHERE ArtistName = 'Ed Sheeran';**

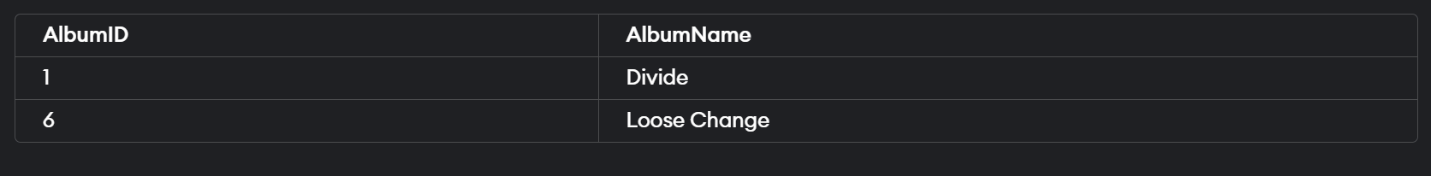


**INSERT INTO album (AlbumID, AlbumName, NumTracks, Genre, ArtistID)  
VALUES (6, 'Loose Change', 8, 'Pop', 1);**



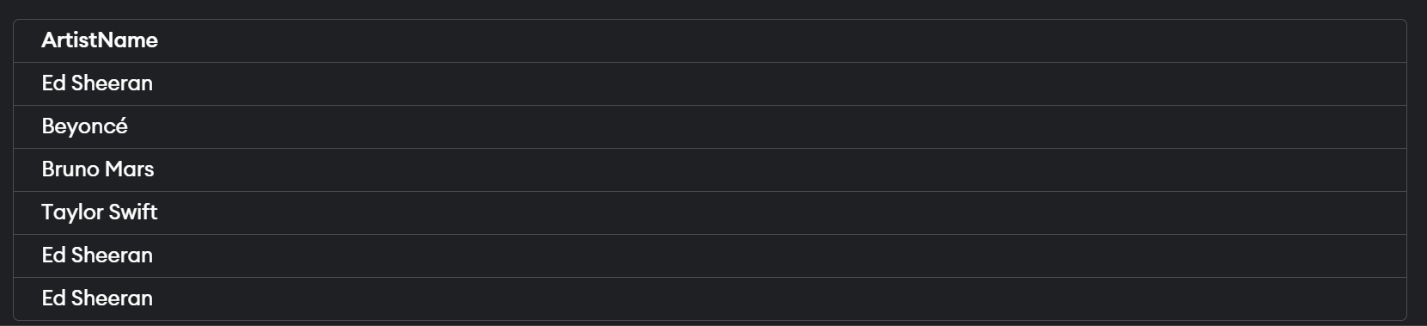
Retrieve the AlbumID and AlbumName of all albums where the ArtistID is 1

**SELECT AlbumID, AlbumName  
FROM album  
WHERE ArtistID = 1;**



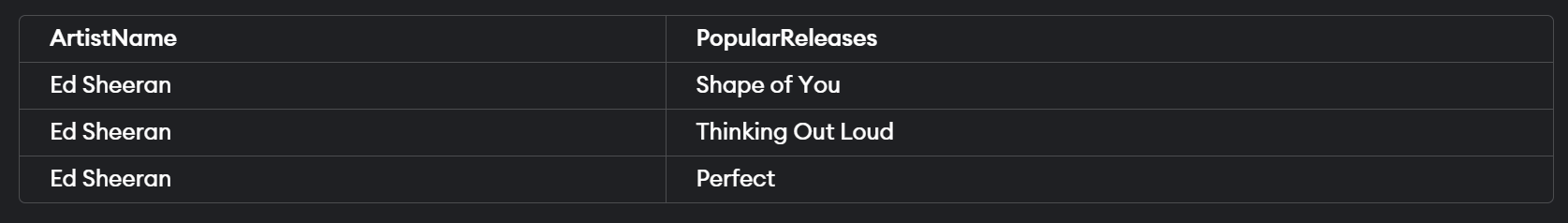
Retrieve the names of artists from the "artist" table whose monthly listeners exceed 3,000,000

**SELECT artistname  
FROM artist  
WHERE MonthlyListeners>3000000;**



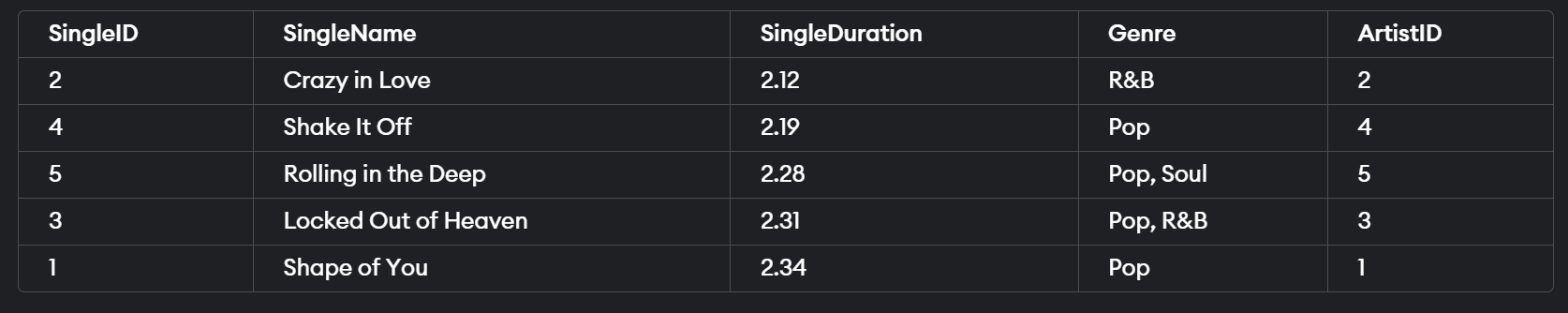
Retrieve the artistname and PopularReleases from the artist table where the artistname starts with 'ed'.

**SELECT artistname, PopularReleases  
FROM artist  
WHERE artistname LIKE 'ed%';**



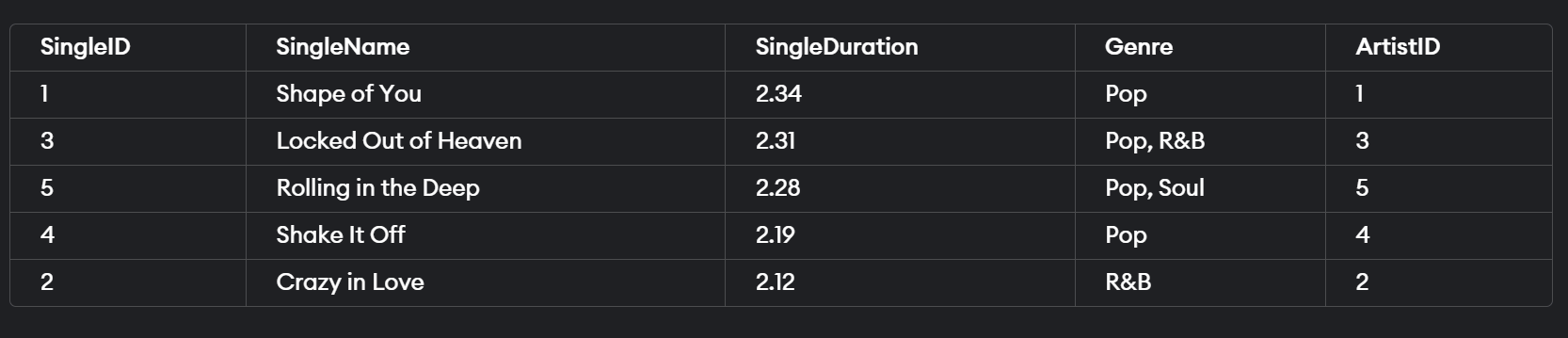
Display the table "Single" in order from the lowest duration to the highest

**SELECT \*  
FROM Single  
ORDER BY SingleDuration ASC;**



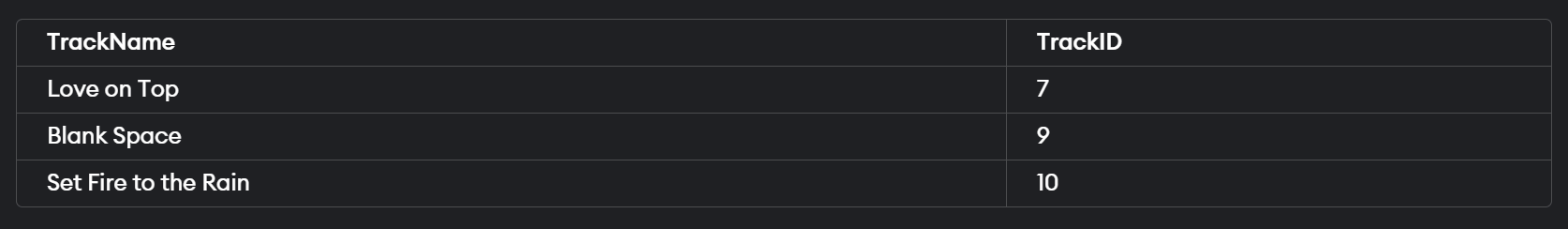
Display the table "Single" in order from the HIGHEST duration to the LOWEST

**SELECT \*  
FROM Single  
ORDER BY SingleDuration DESC;**

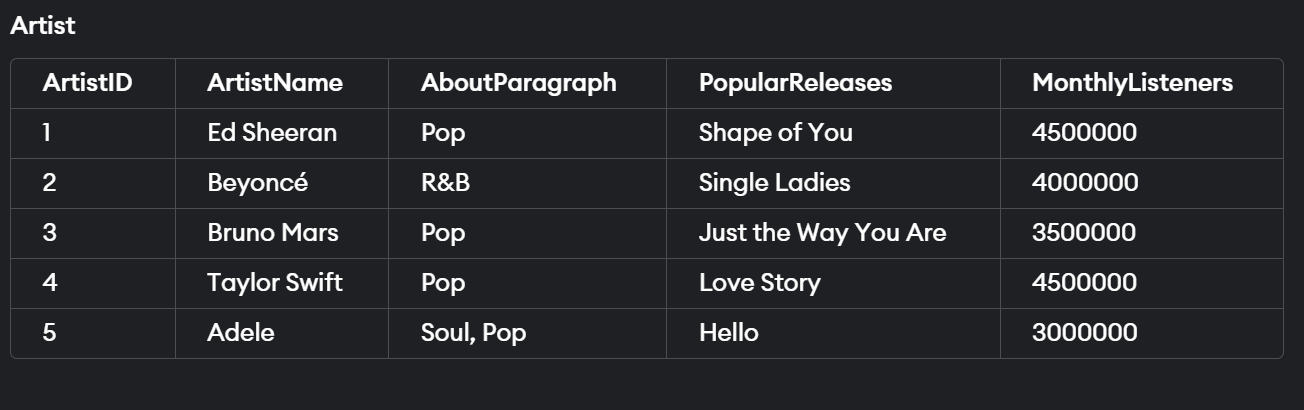


Retrieve the tracks from the "Track" table that have a track duration between 2.1 and 3.

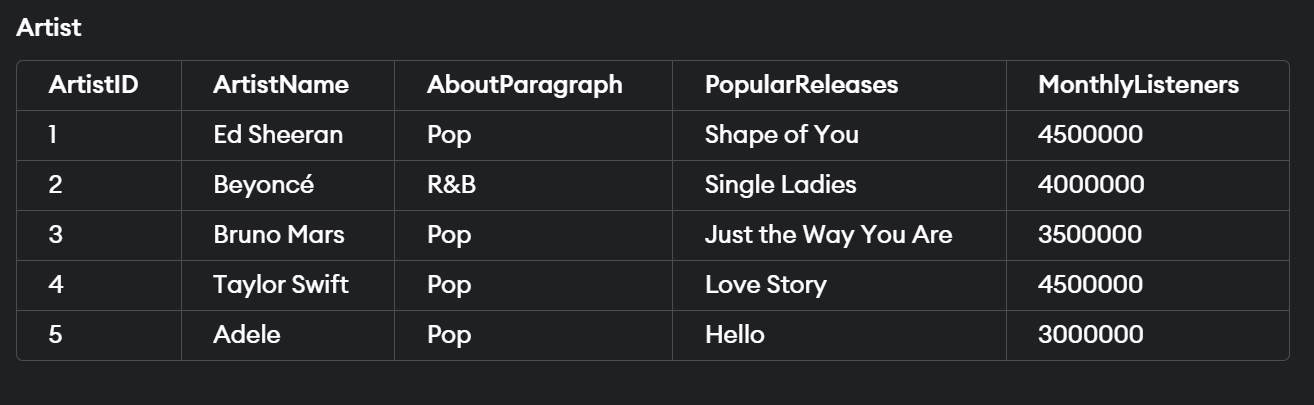
**SELECT TrackName, TrackID   
FROM Track   
WHERE TrackDuration > 2.1 AND TrackDuration < 3;**

****

**DELETE FROM artist WHERE artistid IN (6, 7);**

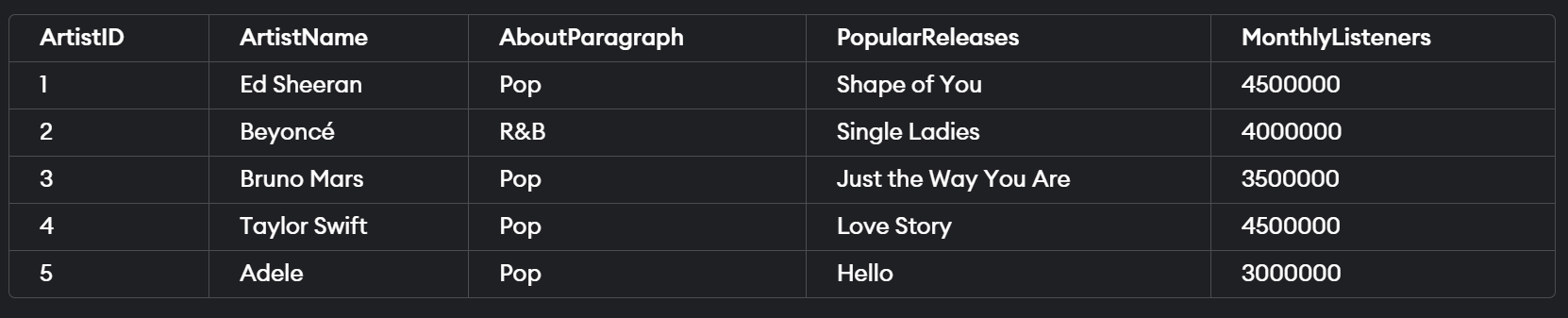
****

**UPDATE ARTIST  
SET AboutParagraph='Pop'   
where artistid=5;**

****

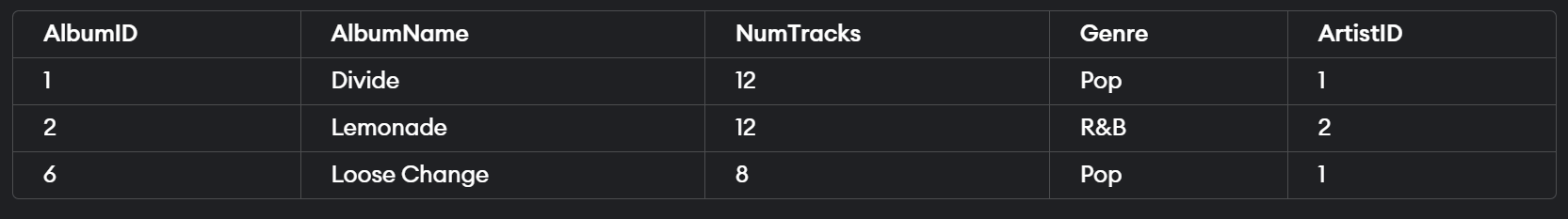
Retrieve all pop and R&B artists

**select \*   
from artist   
where AboutParagraph IN ('Pop' , 'R&B');**

****

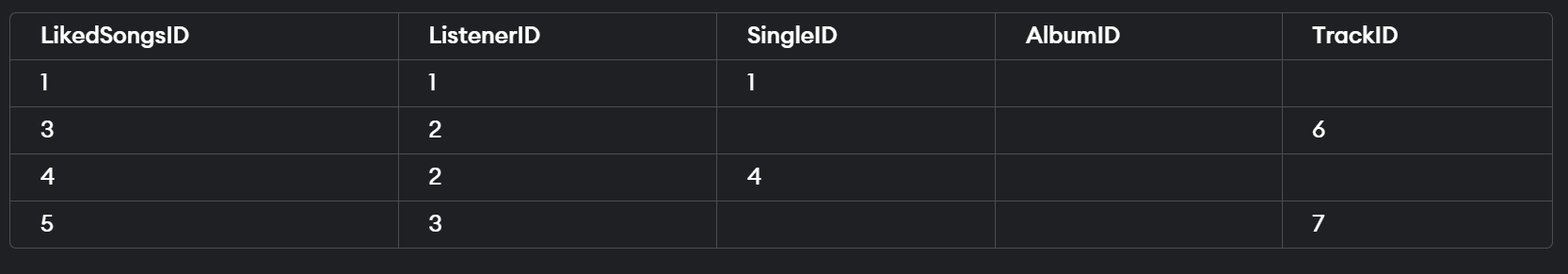
Retrieve all the album names that starts with L or/and ends with e

**SELECT \*  
FROM album  
WHERE albumname LIKE 'L%' OR albumname LIKE '%e';**

****

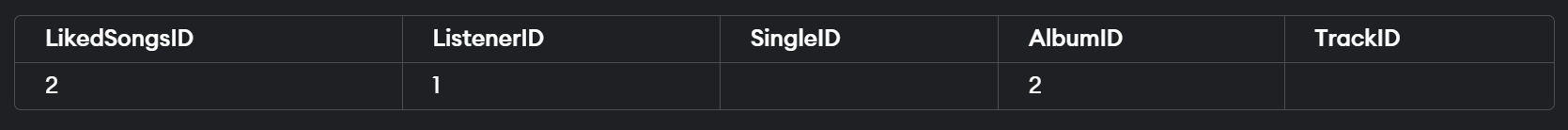
Retrieve all the info where album is null

**SELECT \*  
FROM LikedSongs  
WHERE albumid is null ;**

****

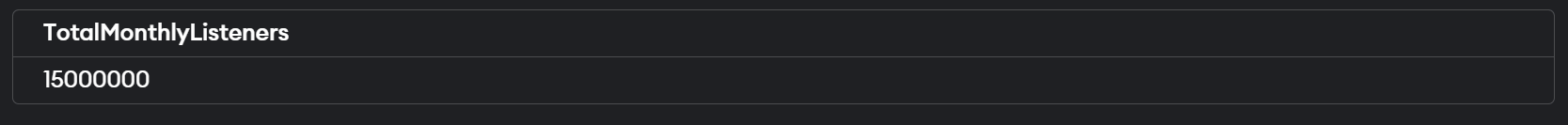
Retrieve all the info where album is not null

**SELECT \*  
FROM LikedSongs  
WHERE albumid is not null ;**

****

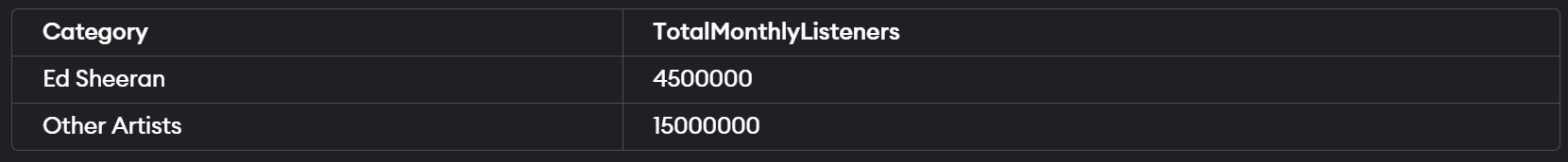
Calculate the sum of MonthlyListeners for all rows in the "Artist" table that have an ArtistName different from 'Ed Sheeran'.

**SELECT SUM(MonthlyListeners) AS TotalMonthlyListeners  
FROM Artist  
WHERE ArtistName <> 'Ed Sheeran';**

****

Calculate the sum of MonthlyListeners for all artists other than Ed Sheeran separately from the sum of MonthlyListeners for Ed Sheeran.

**SELECT  
    CASE  
        WHEN ArtistName <> 'Ed Sheeran' THEN 'Other Artists'  
        ELSE 'Ed Sheeran'  
    END AS Category,  
    SUM(MonthlyListeners) AS TotalMonthlyListeners  
FROM Artist  
GROUP BY category;**

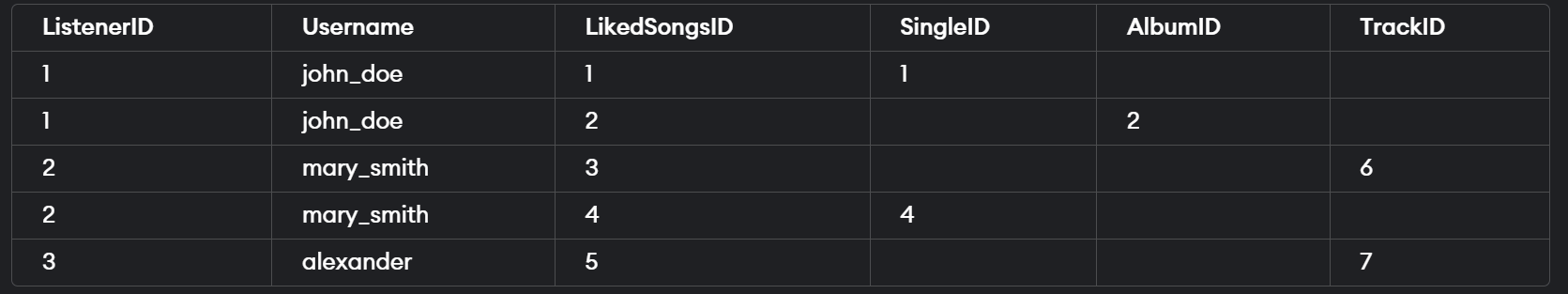
****

Join the "LikedSongs" and "Listener" tables based on the ListenerID column

**SELECT Listener.ListenerID, Listener.Username, LikedSongs.LikedSongsID, LikedSongs.SingleID, LikedSongs.AlbumID, LikedSongs.TrackID  
FROM Listener,liked\_songs**

**where**

**Listener.ListenerID = LikedSongs.ListenerID;**

****