Farideh Tavakoli Giulia Rizzo Mohammad Atabaki

Database Project

Music Streaming Platform

Table of Contents

1- Collect an organic and consistent description of requirements

The main purpose of this work is to establish a database dedicated to storing data associated with a recently introduced worldwide music streaming platform. This platform accommodates users who are distinguished by their unique usernames and possess attributes such as email, password, first name, and last name, among others, and who can have access to different musical tracks. Furthermore, the accessibility to some of the platform functionalities depends on the user's subscription type.

Users are categorized into two types: free users and premium users. Free users are required to view advertisements, each identified by a unique advertisement ID, and the associated company name is stored. The history of visualizations is systematically recorded. Premium users are those who initiate a transaction and pay the annual subscription fee to have access to the download option and exemption from viewing advertisements. Each transaction is recorded. The payment is identified by a unique payment ID and the date, time and payment method information are stored too.

The song track, identified by a unique Track ID and possessing properties like title and duration, should be included in one single/album, which is distinguished by a unique album ID and has attributes such as the release date and the genre. The genre is identified by a unique genre name and each genre has a corresponding description. Users can listen to song tracks, and based on their subscription level they can have access to the download option.

Artists perform songs and can be identified by a unique artist ID; they have an artistic name (which can be either the band name or the individual singer's artistic name) and an optional Instagram link. Each artist is specialized in a particular genre style, and to insert an artist into the platform, the artist should have at least one song, album or single. Moreover, the artist can either be a solo performer or a band, defined when two or more artists collaborate from a certain start date till an optional end date.

The same artist may engage in both solo and collaborative performances, and the respective join date, optionally concluding on an end date, should be stored in a special section. It is important to mention that, in this project, the concept of a band encompasses both one-time or short-term collaborations, where two singers collaborate for a single composition, and long-term collaborations, where two or more singers aim to establish a band in the traditional sense. In our database, we retain information about the most recent collaboration, whether it involves the formation of a band or the creation of a composition.

Users can create or follow playlists, each identified by a unique playlist ID, and they all feature a title, duration and creation date. To be created, playlists must include at least one song track, and the same song track can appear in multiple playlists.

2- Analysis of requirements

2-1- Construction of a glossary of terms

This section presents the glossary of database project. It is designed to clarify the terms used by providing synonyms and a brief description for each. The glossary also highlights logical connections between these terms, aiding in a better understanding of their relationships within the context of the application.

Term	Description	Synonyms	Links
Artist	Who produces and performs tracks and albums	Performer, Creator, Producer, Musician	Band, Singer, Album_Singer, SongTrack, Genre
Band	A group of at least two artists who perform together	Group	Artist, Singer
Singer	An artist who performs alone	Soloist	Artist, Band
Album_Single	One track or more tracks that are recorded and released by the artist with a title	Record, Collection	Artist, SongTrack
SongTrack	A single song recorded by the artist	Music piece	Album_Single, Artist, Genre, Listen_Download_Interaction, Playlist
Genre	A style or category of music	Style, Category	Artist, SongTrack
User	A person who uses the platform to interact with tracks and playlists	Customer, Listener	FreeUser, PremuimUser, Listen_Download_Interaction, Playlist
FreeUser	A user who uses the platform for free	Standard user, Basic user	User, AdvWatching
PremiumUser	A customer who subscribes to a higher tier of service for additional features	Paying member, Pro user	User, Payment
Listen_Download_Interaction	The set of activities a user can do thanks to the service	Engagement	User, SongTrack
Playlist	A list of tracks created by the user	Music collection, Mixtape	SongTrack, User

Advertisement	A notice or announcement in the service promoting a product, event or another service	Commercial	AdvWatching
AdvWatching	The interaction of the free user with the advertisements imposed by the service	Ad viewing	FreeUser, Advertisement
Payment	The action or process of paying for the service	Purchasing	PremiumUser

2-2- Produce a table of operations

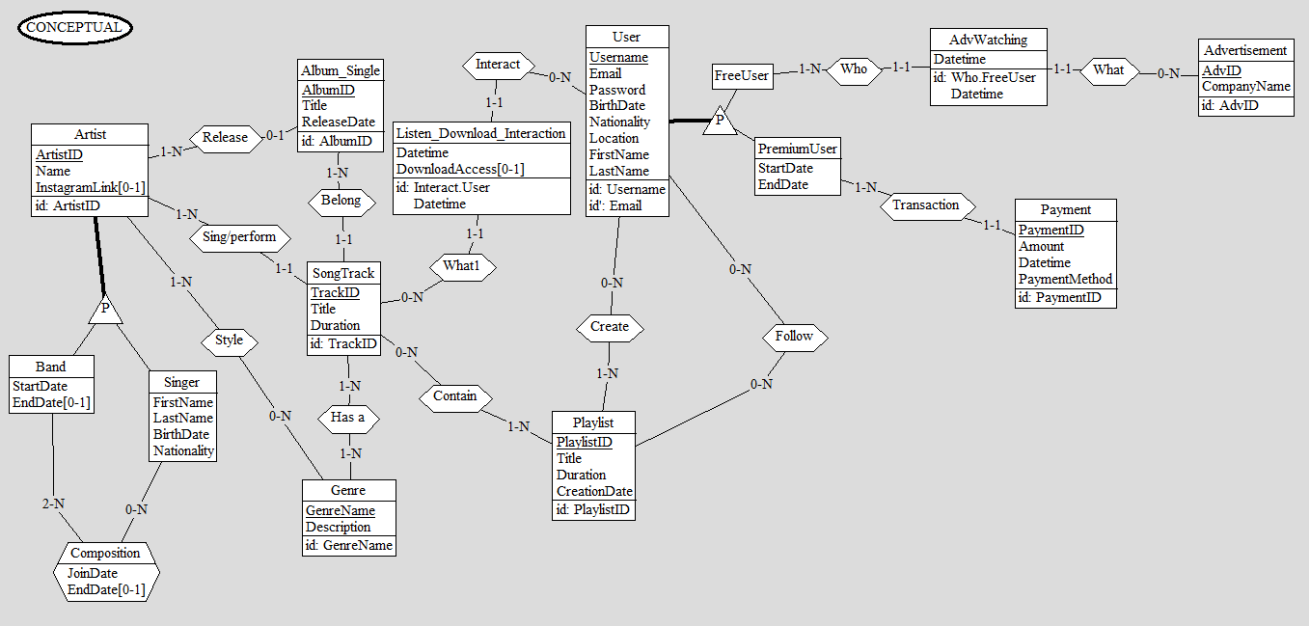
In this project the table of operations is considered only for the main operations, according to the 80-20-rule. The average frequency used in this table refers to the average number of executions over a certain period and the type specified for each operation is either interactive or batch. Interactive operations are the ones related to operations that need real time answers and the batch operations are the ones that are executed often at scheduled intervals.

N°	Operation	Average frequency (Number of times)	Type
1.	Insert a new user	15/Day	Interactive operation
2.	Insert a new song	50/Day	Interactive operation
3.	Insert a new playlist	20/Day	Interactive operation
4.	Insert a new artist	1/Day	Interactive operation
5.	Insert a new advertisement	1/Day	Interactive operation
6.	Count the number of users based on their subscription type	1/Month	Batch operation
7.	Select the artists that perform a particular genre	100/Day	Interactive operation
8.	Select the top 10 trending songs listened to and downloaded by users	1/Week	Batch operation
9.	Count the number of playlists containing the most recently released track	1/Week	Batch operation
10.	Find the albums' titles released by an artist	100/Day	Interactive operation
11.	Find the songs with a specific title in the search bar	500/Day	Interactive operation

12.	Find the average number of advertisements' visualizations, a user does per month	1/Month	Batch operation
13.	Count the number of downloads of a song track	1/Month	Batch operation
14.	Count the number of followers of a specific playlist	1/Week	Batch operation

3- Conceptual design

3-1- Design an E/R model



Derived attributes / non-modellable constraints:

- An advertisement is shown to a free user after listening to 3 songs.
- The only option for the subscription duration is 1 year with a 1-month free trial.

3-2- Produce a table of volumes

Let's consider an analysis of the volumes one year after the platform's initial launch.

Concept	Type	Volume
Artist	**E**	**5000**
Release (Artist → Album/Single)	R	Artist*1 (on avg)
Album_Single	**E**	**1*5000 (year)**
Sing/Perform (Artist → Song/Track)	R	Artist* 10 (on avg)

Belong (Song/Track → Album/Single)	R	Artist * 10 * 1
SongTrack	**E**	**10*5000(year)**
Band	**E**	**2000**
Singer	**E**	**4000**
Composition (Singer → Band)	R	1000 (on avg) 30% (3000)
Genre	**E**	**50**
Style (Artist → Genre)	R	Artist*3 (on avg)
Has a (SongTrack → Genre)	R	Song/Track * 3 (on avg)
User	**E**	**50.000**
Create (User → Playlist)	R	Users*5 (on avg)
Follow (User → Playlist)	R	Users*10
Playlist	**E**	**5*50.000**
Contain (Playlist → songTrack)	R	Playlist*20 (on avg)
Listen/Download/Interaction	**E**	**10*365*users**
Interact (User → Listen/Download/Interaction)	R	10(on avg)*365*users
What1 (Listen/Download/Interaction → SongTrack)	R	10(on avg) * 365*Song/Tracks
FreeUser	**E**	**35.000**
AdvWatching	**E**	**10*365*35000/3 = 42.583.000**
Who (AdvWatching → Free user)	R	Free users
What (AdvWatching → Advertisement)	R	Advertisement
Advertisement	**E**	**1000**
PremiumUser	**E**	**15.000**
Transaction (PremiumUser → Payment)	R	15.000
Payment	**E**	**15.000**

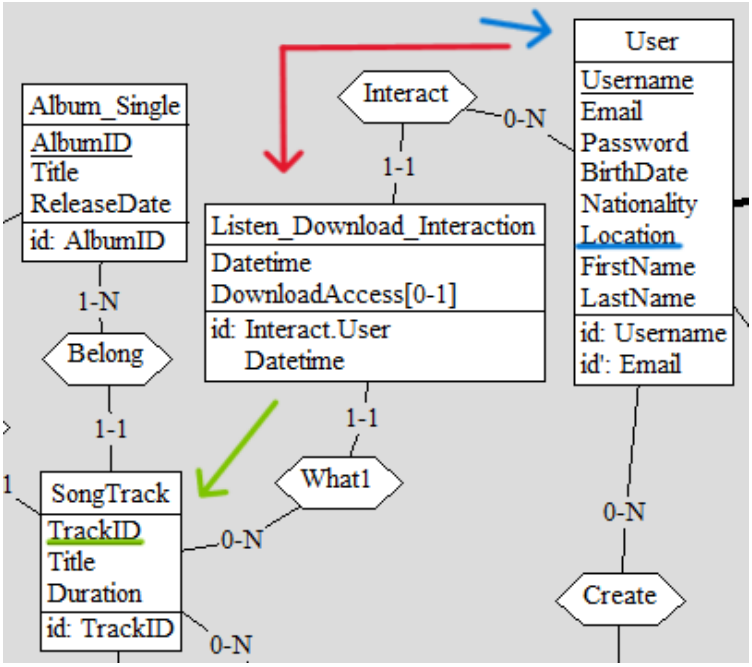
3-3- Cost assessment (only for 1 operation)

We decided to work on operation 8.
Operation 8: Select the top 10 trending **songs** **listened to and downloaded** by **users** in a specific country.

3-3-1- Navigation schema

Let's suppose we want to find the top 10 trending songs listened to or downloaded by users in the UK.
We start our analysis from the "User" entity, which includes the "location" attribute, essential to identify UK-based users. This entity is linked to the "Listen_Download_Interaction" entity through the "interact" relationship, and from this entity we can retrieve all the listening and downloading interactions. Then, we move to the last entity required, "SongTrack", through the "what1" relationship, and sort the first 10 TrackIDs with the highest interactions count in descending order.

The navigation schema depicted in the following picture represents the fragment of the E/R schema relevant to this operation. It efficiently retrieves the top 10 trending songs listened to or downloaded by users from the UK.



3-3-2- Table of accesses

Let's suppose that we have identified 5000 users of the platform located in the UK; additionally, we had previously estimated an average of **10 interactions per user** per day.

Operation number 8. can be summarized in the following access table:

Concept	Type	Accesses	Type
User	E	1	R
Interact	R	10	R
Listen_Download_Interaction	E	10	R
What1	R	10	R
SongTrack	E	10	R
TOTAL	-	41	-

1. We need 1 access to the User table to select all the users from the UK;
2. 10 accesses are required to link the User table to the Listen_Download_Interaction table, to associate the British users with each of their listening and downloads;
3. 10 accesses to the Listen_Download_Interaction table are needed to visualize the relevant listen/download interactions;

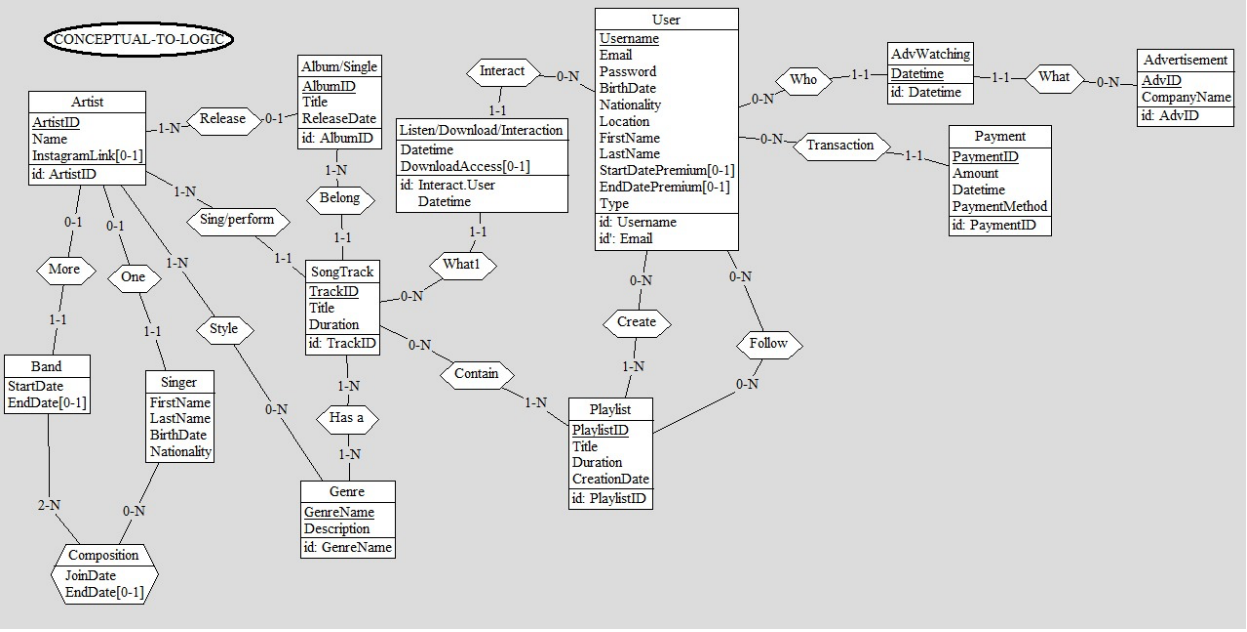
4. Again, 10 accesses are needed to link the Listen_Download_Interaction table to the SongTrack table, to associate the interactions we are interested in with each song track IDs;
5. Finally, with 10 other accesses we can sort the 10 song tracks with the highest number of interactions;
6. By summing all the accesses, we obtain a total of 41 accesses. All these accesses are *reading* accesses.

4- Logical design

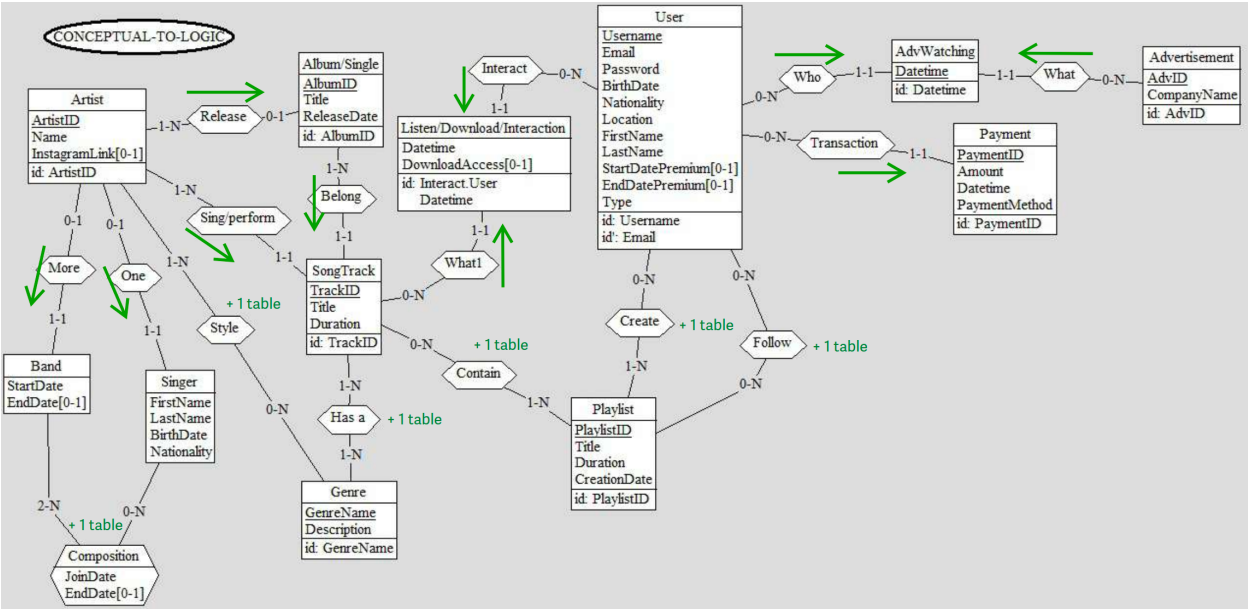
4-1- Conceptual-to-Logic model

4-1-1- Restructuring phase

The original schema included two generalizations (**Band-Singer** from the parent Artist and **FreeUser-PremiumUser** from the parent User); we removed the former by replacing it with new relationships, and the latter by collapsing upwards and adding a new attribute, "Type".
The schema didn't include any multivalued attributes; therefore, no further restructuring was necessary.



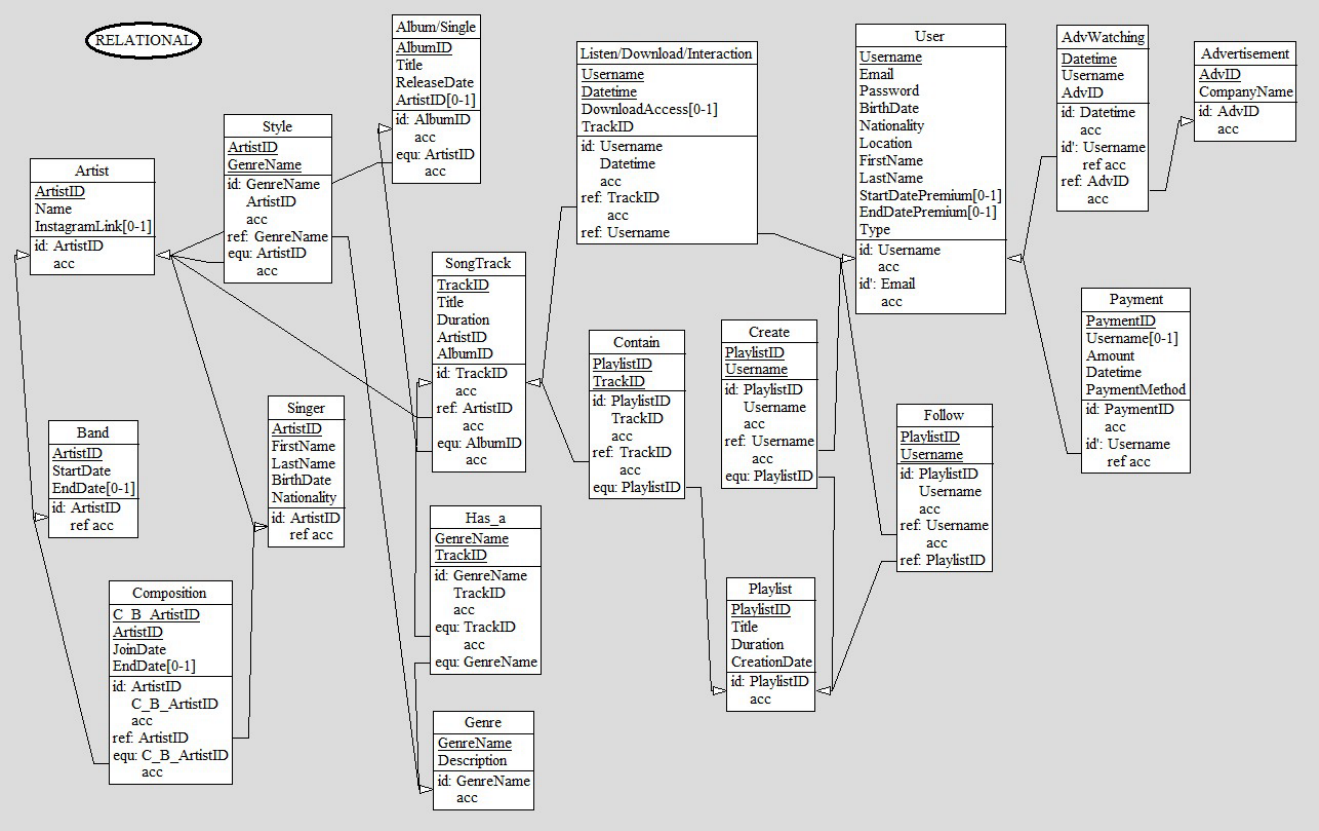
4-1-2- Translation phase



1. **Artist** (ArtistID, Name, InstagramLink*)
2. **Band** (ArtistID: Artist, StartDate, EndDate*)
3. **Singer** (ArtistID: Artist, FirstName, LastName, BirthDate, Nationality)
4. **Composition** (band: Band, singer: Singer, JoinDate, EndDate*)
5. **AlbumSingle** (AlbumID, Title, ReleaseDate, ArtistID*: Artist)
6. **SongTrack** (TrackID, Title, Duration, AlbumID: Album/Single, ArtistID: Artist)
7. **Genre** (GenreName, Description)
8. **Has a** (GenreName: Genre, TrackID: SongTrack)
9. **Style** (GenreName: Genre, ArtistID: Artist)
10. **User** (Username, **AK**: Email, Password, BirthDate, Nationality, Location, FirstName, LastName, StartDatePremium*, EndDatePremium*, Type)
11. **ListenDownloadInteraction** (DateTime, Username: User, DownloadAccess*, TrackID: SongTrack)
12. **PlayList** (PlaylistID, Title, Duration, CreationDate)
13. **Contain** (TrackID: SongTrack, PlayListID: Playlist)
14. **Create** (PlaylistID: Playlist, Username: User)
15. **Follow** (PlaylistID: Playlist, Username: User)
16. **AdvWatching** (DateTime, Username: User, AdvID: Advertisment)
17. **Advertisment** (AdvID, CompanyName)
18. **Payment** (PaymentID, Amount, DateTime, PaymentMethod, Username: User)

4-2- Relational model

4-2-1- Relational schema



5- SQL queries

The number of the queries corresponds to the number of the table of operation.

1. Insert a new user

INSERT INTO music_streaming_platform.User (Username, Email, Password, Birthdate, Nationality, Location, FirstName, LastName, StartDatePremium, EndDatePremium, Type)
VALUES ('Simo_Silvi','simone.silvi33@gmail.com,','sisi5769','1999-09-16', 'Italian', 'Milan', 'Simone', 'Silvi', null, null, 'Free');

6. Count the number of the users based on their subscription type

SELECT COUNT (Username) FROM music_streaming_platform.User
WHERE Type = 'Premium';

SELECT COUNT (Username) FROM music_streaming_platform.User
WHERE Type = 'Free';

7. Select the artists that perform a particular genre

```
SELECT * FROM music_streaming_platform.artist
JOIN music_streaming_platform.style
ON artist.ArtistID = style.ArtistID
WHERE style.GenreName = 'Metal';
```

8. Select the top 10 trending songs listened to and downloaded by users

```
SELECT songtrack.TrackID, Title, Duration, COUNT (listen_download_interaction.TrackID) AS
InteractionCount FROM music_streaming_platform.songtrack
JOIN music_streaming_platform.listen_download_interaction
ON songtrack.TrackID = listen_download_interaction.TrackID
GROUP BY songtrack.TrackID, Title, Duration
ORDER BY InteractionCount desc
LIMIT 10;
```

10. Find the albums' titles released by an artist name.

```
SELECT Title, Name FROM music_streaming_platform.album_single
JOIN music_streaming_platform.artist
ON artist.ArtistID = album_single.ArtistID;
```

11. Find the songs with a specific title in the search bar.

```
SELECT * FROM music_streaming_platform.songtrack
WHERE Title = 'Creepin';

SELECT * FROM music_streaming_platform.songtrack
WHERE Title = 'Numb';
```

14. Count the number of followers of a specific playlist.

```
SELECT COUNT (Username) FROM music_streaming_platform.follow
JOIN music_streaming_platform.playlist
ON follow.PlaylistID = playlist.PlaylistID
WHERE Title = 'Metal Mix';
```