**Spotify Database Project**

**About Company:**

The Spotify database serves as the foundation of one of the world's premier music streaming platforms, connecting millions of users with a vast library of music from across the globe. At its core, the database is structured around several key entities that enable users to seamlessly navigate and interact with the platform's extensive collection of music content.

Central to the Spotify experience are user profiles, which store individual preferences, interaction history, and personalized settings. Through their profiles, users can curate their music listening experience, create playlists, follow favorite artists, and engage with other members of the Spotify community. Artists and albums are meticulously cataloged within the database, providing users with rich metadata and detailed insights into their favorite musicians and their discographies.

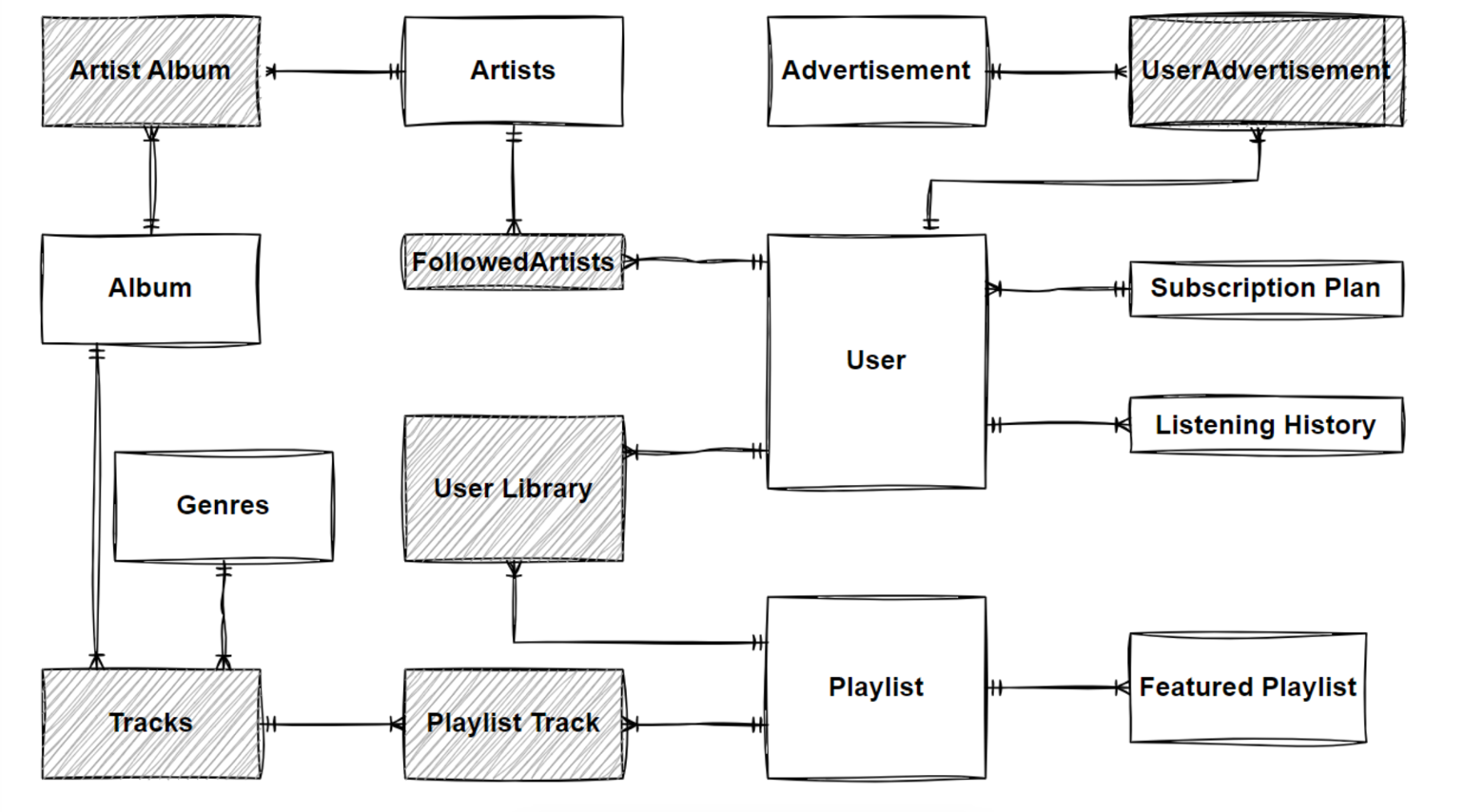
The heart of the Spotify database lies in its extensive collection of tracks, each meticulously indexed with metadata such as titles, durations, genres, and popularity ratings. This vast repository of music content forms the basis for Spotify's sophisticated recommendation algorithms, which leverage user preferences, listening history, and behavioral data to deliver personalized music recommendations tailored to each individual user.

Besides personalized recommendations, Spotify lets users explore new music through handpicked playlists, highlighted albums, and browsing by genre. These carefully selected options, powered by the Spotify database, offer users various ways to discover music, ensuring there’s always something fresh to enjoy.

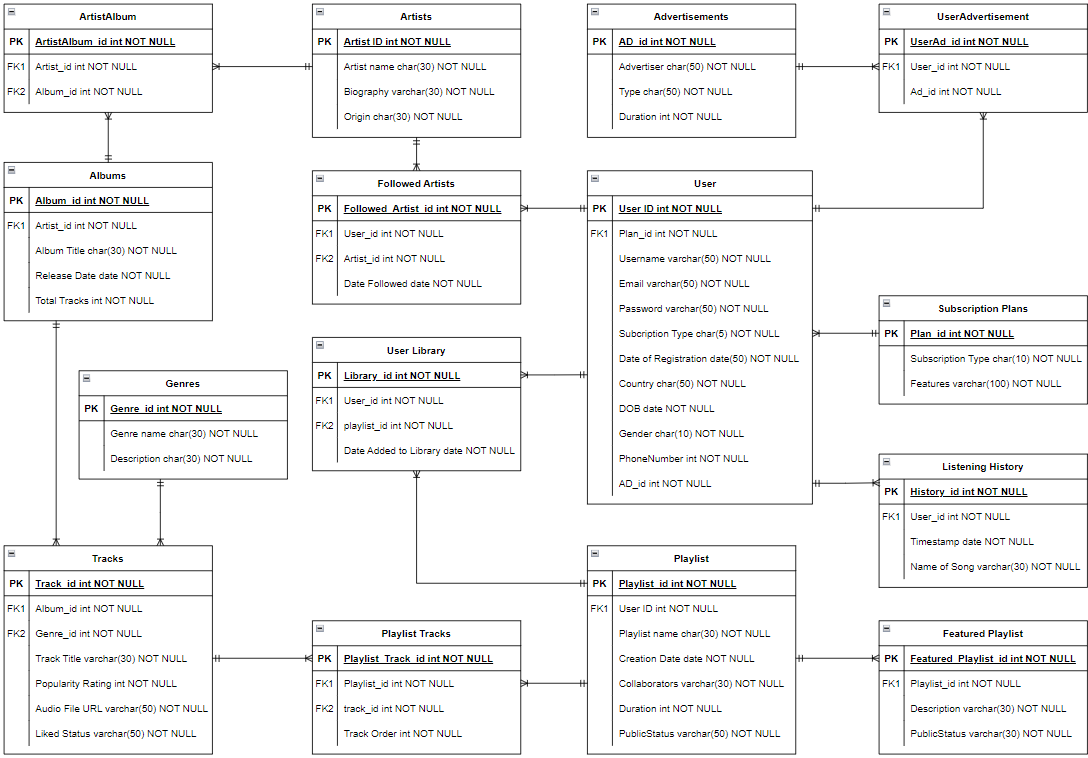
Spotify’s database not only helps you find and recommend music, but also targets ads similarly to things you may like. This is another way that Spotify can make revenue as an app and keep updating itself to fit users’ interests. In addition, Spotify offers different subscriptions for customers. For example, Spotify offers many different premium plans which include solo, duo, family, and students. All plans have different pricing and may be more ideal for a person or family depending on the current situation.

Spotify manipulates their data to create a social platform on their music application, their database sets the stage for a better music streaming experience, providing personalized recommendations, curated playlists, and ways to make money. It also allows users to share with each other and listen to one another’s playlists. With smart algorithms and organized data, Spotify aims to keep users happy and the business growing strong.

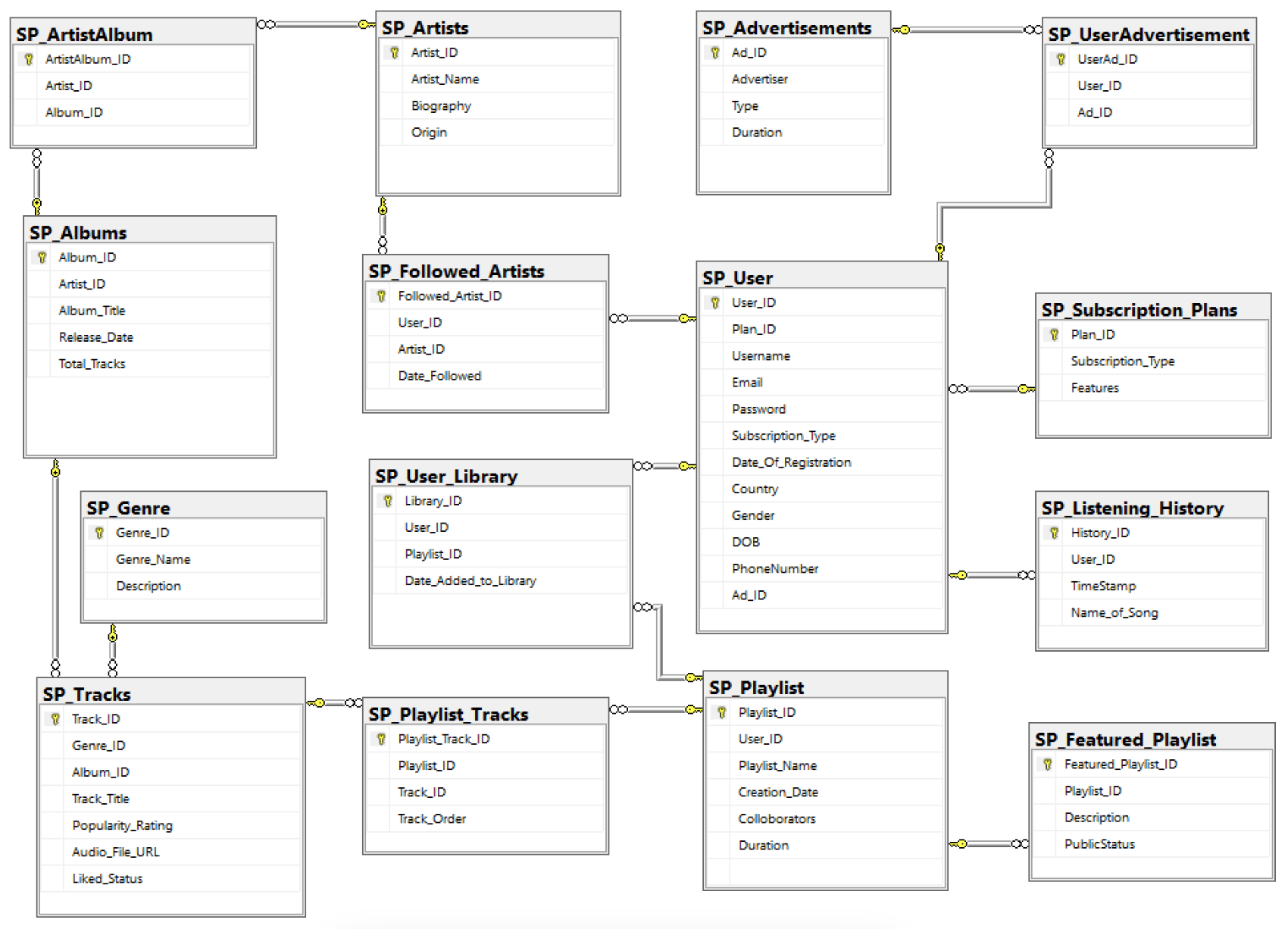
**ER – Diagram:**



**ER Diagram with attributes:**



**Database Design with list of tables:**



**QUERIES**:

--1) Who are all having spotify premium accounts ?

select Username, Subscription\_Type, Gender

from SP\_User

where Plan\_ID in ('1','2','3','4')

order by Subscription\_Type

--2) Customers who are targeted by business commercials

select\* from SP\_User where Ad\_ID in

(select Ad\_ID from SP\_UserAdvertisement where Ad\_ID in

(select Ad\_ID

from SP\_Advertisements where Type='Insurance'))

--3) Display Drake's albums.

select Artist\_Name, Album\_Title, Release\_Date, Total\_Tracks

from SP\_Albums SA, SP\_ArtistAlbum SAA, SP\_Artists SAAA

where SA.Artist\_ID=SAA.Artist\_ID and SAA.Artist\_ID=SAAA.Artist\_ID and Artist\_Name='Drake'

--4) Show all SPOTIFY  featured playlist

 select Playlist\_Name from SP\_Featured\_Playlist where playlist\_ID in

(select Playlist\_ID from SP\_Playlist where USER\_ID in

(select USER\_ID from SP\_User\_Library))

--5) What are the top rated songs in spotify?

select Genre\_Name,Track\_Title,Description, Popularity\_Rating

from SP\_Genre SG, SP\_Tracks ST

where SG.Genre\_ID=ST.Genre\_ID and Popularity\_Rating>='8'

order by Popularity\_Rating desc

--6) Find the number of tracks in an album id 1100

SELECT t.Track\_Title

FROM SP\_Albums a

JOIN SP\_Tracks t ON a.Album\_ID = t.Album\_ID

where t.Album\_ID='1100'

--7) Display which album have the genre type R&B

SELECT DISTINCT a.Album\_ID, a.Album\_Title

FROM SP\_Albums a

JOIN SP\_Tracks t ON a.Album\_ID = t.Album\_ID

JOIN SP\_Genre g ON t.Genre\_ID = g.Genre\_ID

WHERE g.Genre\_Name = 'R&B';

--8) Dispaly the total number of playlists created by each user

SELECT SP\_user.username, COUNT(SP\_Playlist.Playlist\_ID) AS Total\_Playlists

FROM SP\_user

LEFT JOIN SP\_Playlist ON SP\_user.user\_id = SP\_Playlist.User\_ID

GROUP BY SP\_user.username

--9) Display users who have followed more than 5 artists.

SELECT SP\_user.username, COUNT(SP\_Followed\_Artists.Artist\_ID) AS Total\_Followed

FROM SP\_user

LEFT JOIN SP\_Followed\_Artists ON SP\_user.user\_id = SP\_Followed\_Artists.User\_ID

GROUP BY SP\_user.username

HAVING COUNT(SP\_Followed\_Artists.Artist\_ID) >= 1

--10) Display users who have subscribed to the 'Duo' plan and what features does they get

SELECT Username, Features

FROM SP\_user s , SP\_Subscription\_Plans p

WHERE s.Subscription\_Type=s.Subscription\_Type and p.Subscription\_Type='Duo'

--11) List all tracks with a popularity rating greater than 8 and its geners

SELECT Track\_Title, Popularity\_Rating

FROM SP\_Tracks t, SP\_Genre g

WHERE t.Genre\_ID=g.Genre\_ID and  Popularity\_Rating > 8

--12) Count the number of tracks in each genre:

 SELECT Genre\_Name, COUNT(Track\_ID) AS track\_count

FROM SP\_Tracks t, SP\_Genre g

where t.Genre\_ID=g.Genre\_ID

GROUP BY Genre\_Name

--13) Display the names of all artists and the titles of their albums.

SELECT A.Artist\_Name, AL.Album\_Title

FROM SP\_Artists A,SP\_ArtistAlbum AA ,SP\_Albums Al

where A.Artist\_ID=AA.Artist\_ID AND AA.Album\_ID=Al.Album\_ID;

--14) Display the names of all artists followed by the user with the ID '800'.

SELECT Artist\_Name

FROM SP\_Artists

WHERE Artist\_ID IN (

SELECT Artist\_ID

FROM SP\_Followed\_Artists

WHERE Followed\_Artist\_ID = 800);

--15) Display username, email, and phone number of users who have a student subscription plan

SELECT u.Username, u.Email, u.PhoneNumber

FROM SP\_User u

JOIN SP\_Subscription\_Plans s ON u.Plan\_ID = s.Plan\_ID

WHERE s.Subscription\_Type = 'Student';

-- 16) Display which ad targeted which user

SELECT U.Username, A.Advertiser

FROM  SP\_User U,  SP\_UserAdvertisement UA, SP\_Advertisements A

where UA.User\_ID = U.User\_ID and  UA.Ad\_ID = A.Ad\_ID;

--17) User gender and subscription plan

SELECT  U.Gender, S.Subscription\_Type

FROM SP\_User U

JOIN SP\_Subscription\_Plans S ON U.Plan\_ID = s.Plan\_ID

order by Gender

--18) Display all the user playlist in the library along with the created date.

Select Playlist\_Name, Colloborators,Creation\_Date

from SP\_User\_Library A, SP\_Playlist B

where A.Playlist\_ID=B.Playlist\_ID

--19) List all the songs of user '100' from user history

select Username, Name\_of\_Song

from SP\_User s, SP\_Listening\_History h

where s.User\_ID = h.User\_ID and s.User\_ID='100'

--20) Display all users who have not added any tracks to their library.

SELECT u.username

FROM SP\_user u

LEFT JOIN SP\_User\_Library ul ON u.user\_id = ul.User\_ID

WHERE ul.Library\_ID IS NULL