**SPOTIFY ADVANCE SQL ANALYSIS**

SELECT TOP (1000) [Artist]

,[Track]

,[Album]

,[Album\_type]

,[Danceability]

,[Energy]

,[Loudness]

,[Speechiness]

,[Acousticness]

,[Instrumentalness]

,[Liveness]

,[Valence]

,[Tempo]

,[Duration\_min]

,[Title]

,[Channel]

,[Views]

,[Likes]

,[Comments]

,[Licensed]

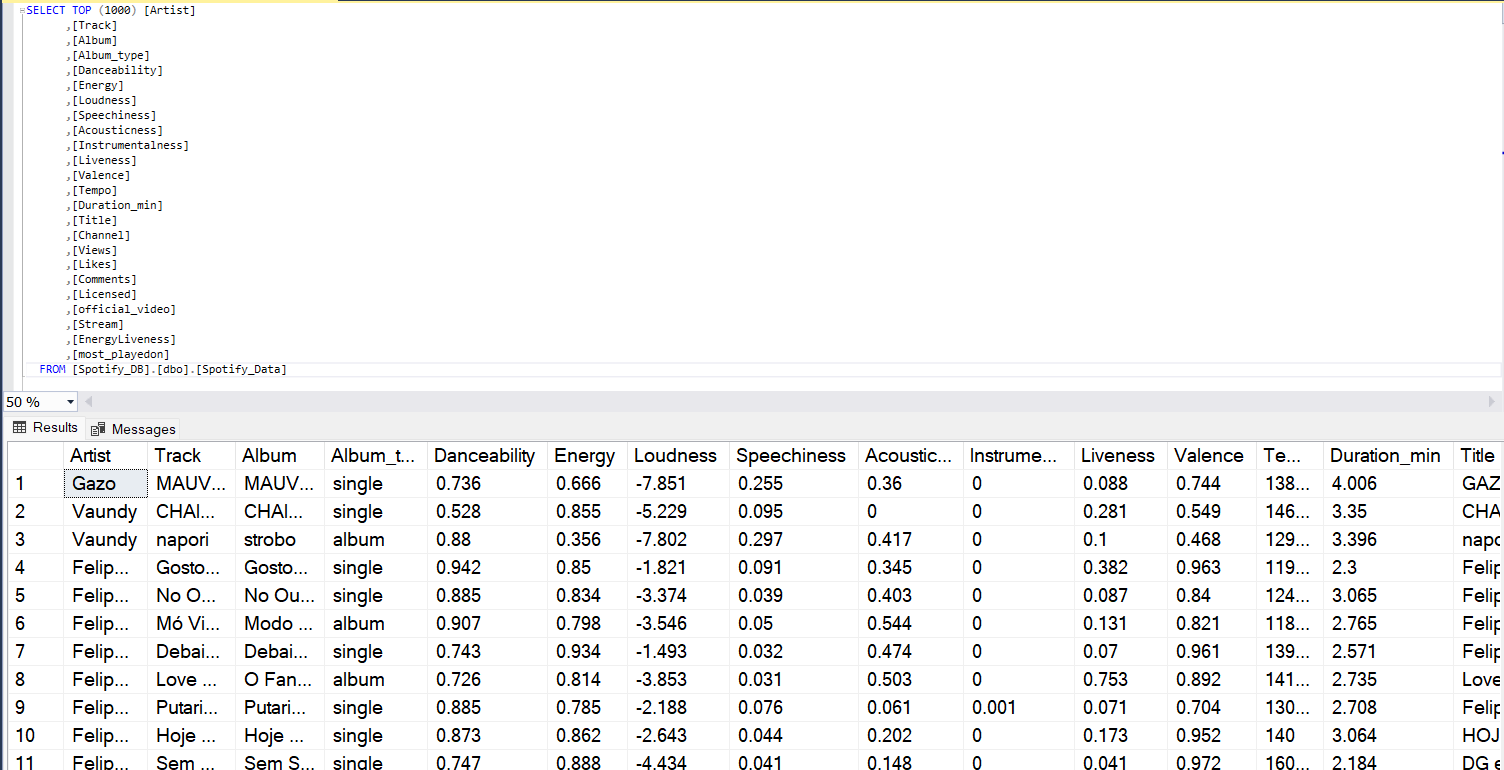
,[official\_video]

,[Stream]

,[EnergyLiveness]

,[most\_playedon]

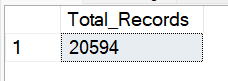
FROM [Spotify\_DB].[dbo].[Spotify\_Dataset]



**EXPLORATORY DATA ANALYSIS**

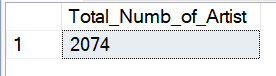
How many records are there?

SELECT COUNT(\*) Total\_Records FROM Spotify\_Data



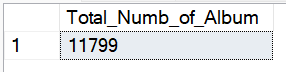
What is the Total Number of Unique Artist we have?

SELECT COUNT(DISTINCT Artist) Total\_Numb\_of\_Artist FROM Spotify\_Data



What is the Total Number of Albums?

SELECT COUNT(DISTINCT Album) Total\_Numb\_of\_Album FROM Spotify\_Data



What are the Album type available?

SELECT DISTINCT Album\_type FROM Spotify\_Data



What is the Maximum and Minimum Duration of a Song?

SELECT

MAX(Duration\_min) Max\_Duration,

MIN(Duration\_min) Min\_Duration

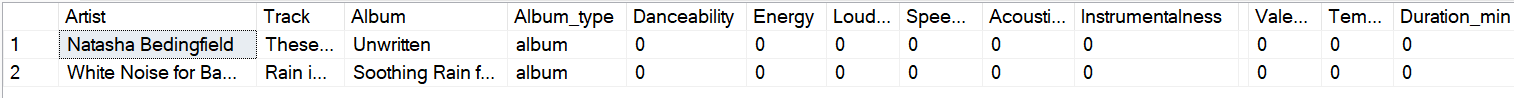
From Spotify\_Data



Some songs are having 0 duration that cannot be possible so we need to get rid of those errors

SELECT \* FROM Spotify\_Data

WHERE Duration\_min = 0



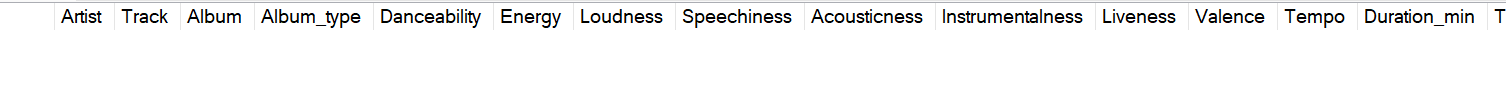
So we will Delete these two records

DELETE FROM Spotify\_Data

WHERE Duration\_min = 0

SELECT \* FROM Spotify\_Data

WHERE Duration\_min = 0



Now How many records are there?

SELECT COUNT(\*) Total\_Records FROM Spotify\_Data



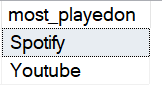
How many channel we have?

SELECT COUNT(DISTINCT Channel) Number\_of\_Channels FROM Spotify\_Data



What are the Platform in which we segment most popular

SELECT DISTINCT most\_playedon FROM Spotify\_Data



===================================================

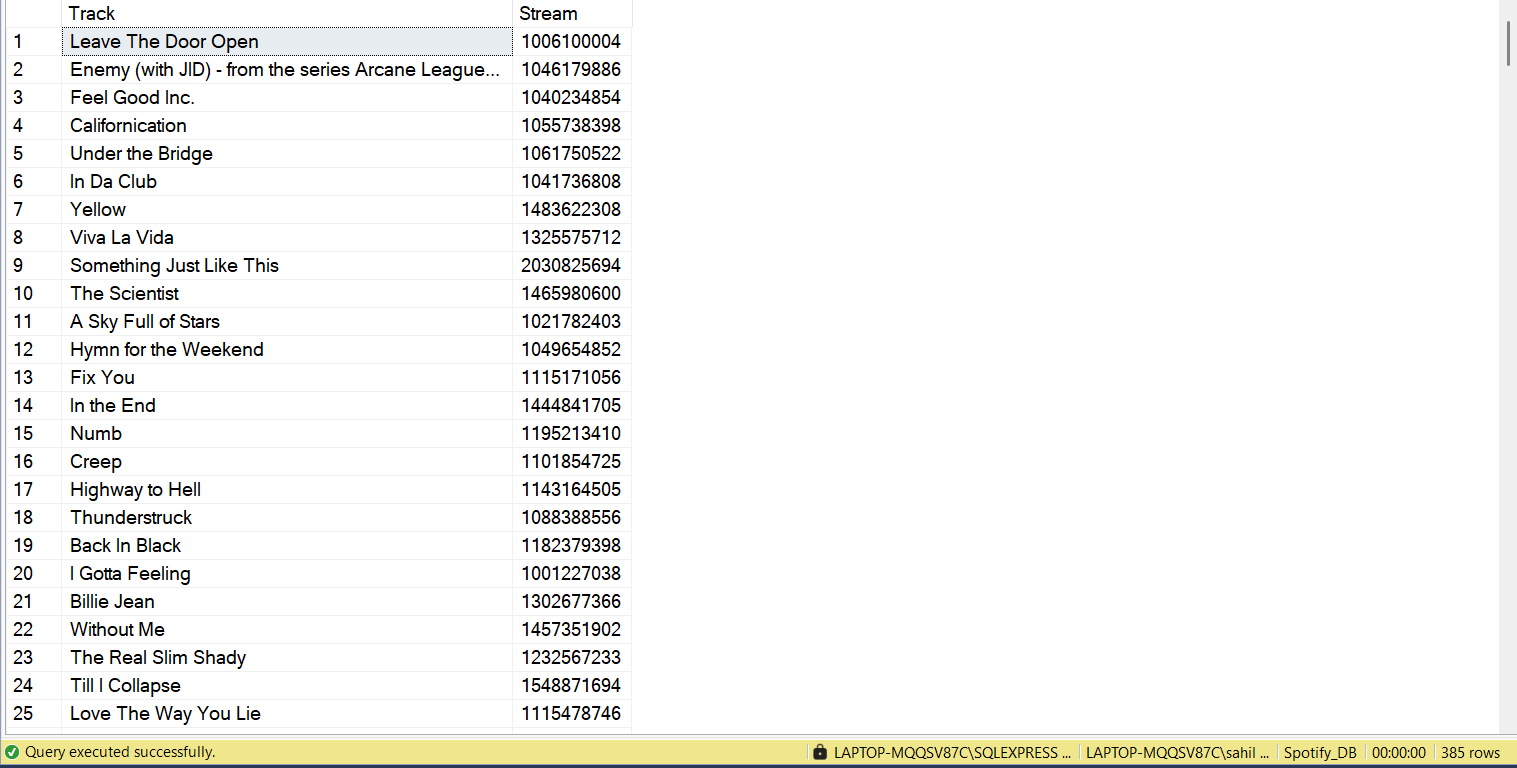
**Data Analysis Basic level**

**==**=================================================

(Q1) Retrieve the names of all tracks that have more than 1 billion stream

SELECT Track, Stream FROM Spotify\_Data

WHERE Stream > 1000000000

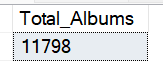


(Q2) List all the albums with their respective artists

SELECT

COUNT(DISTINCT Album) Total\_Albums

FROM Spotify\_Data



Here one thing to Conlude that More than One Artist made the Album

SELECT

DISTINCT Album,

Artist

FROM Spotify\_Data

ORDER BY Album



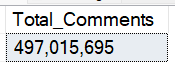
(Q3) Get the total number of comments for tracks where licensed = True

SELECT

FORMAT((SUM(Comments)) ,'N0', 'en-US') Total\_Comments

FROM Spotify\_Data

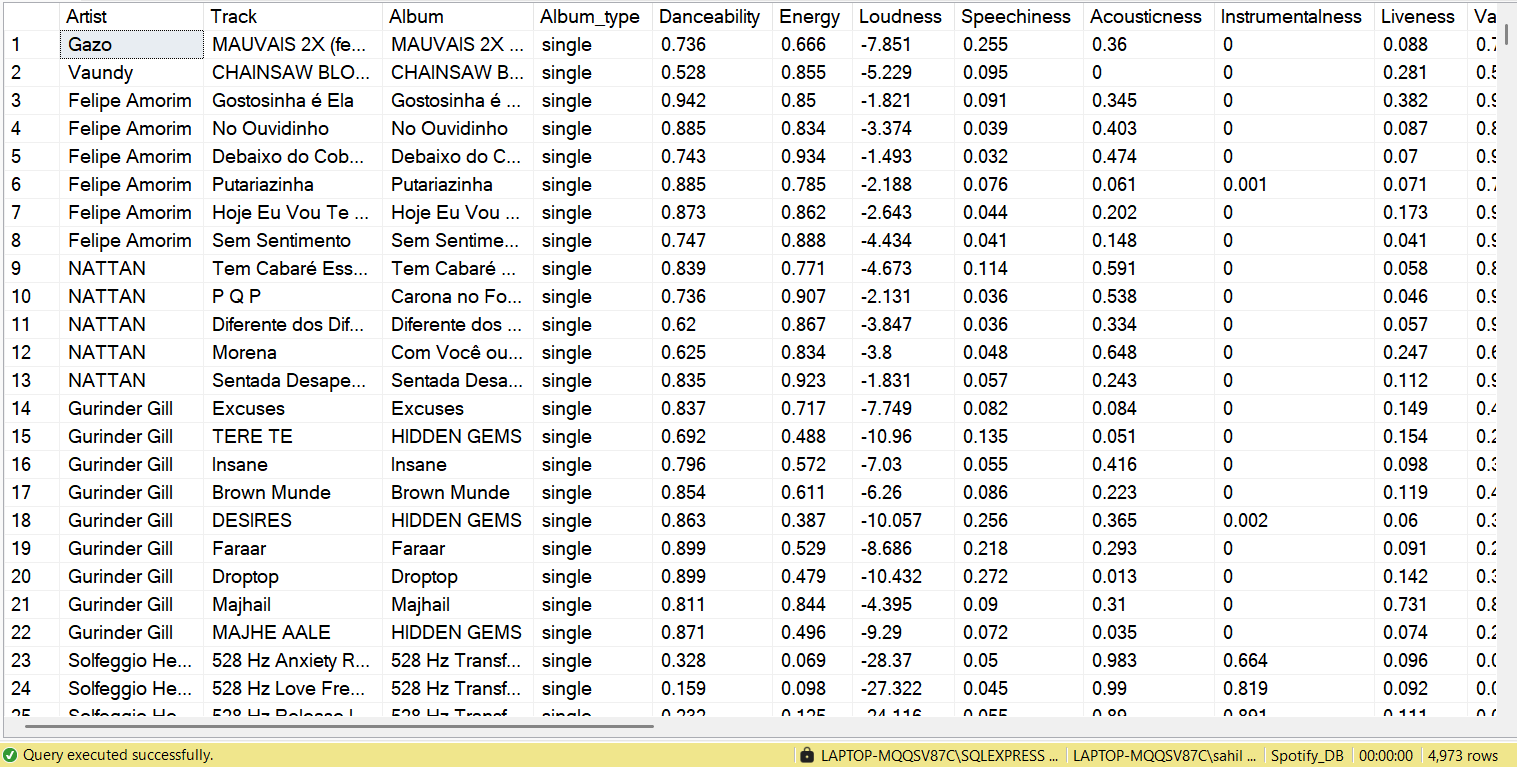
WHERE Licensed = 'TRUE'



(Q4) Find all the Track that belong to album type single.

SELECT \* FROM Spotify\_Data

WHERE Album\_type = 'Single'



(Q5) Count the Total Number of tracks by each artist

SELECT

Artist,

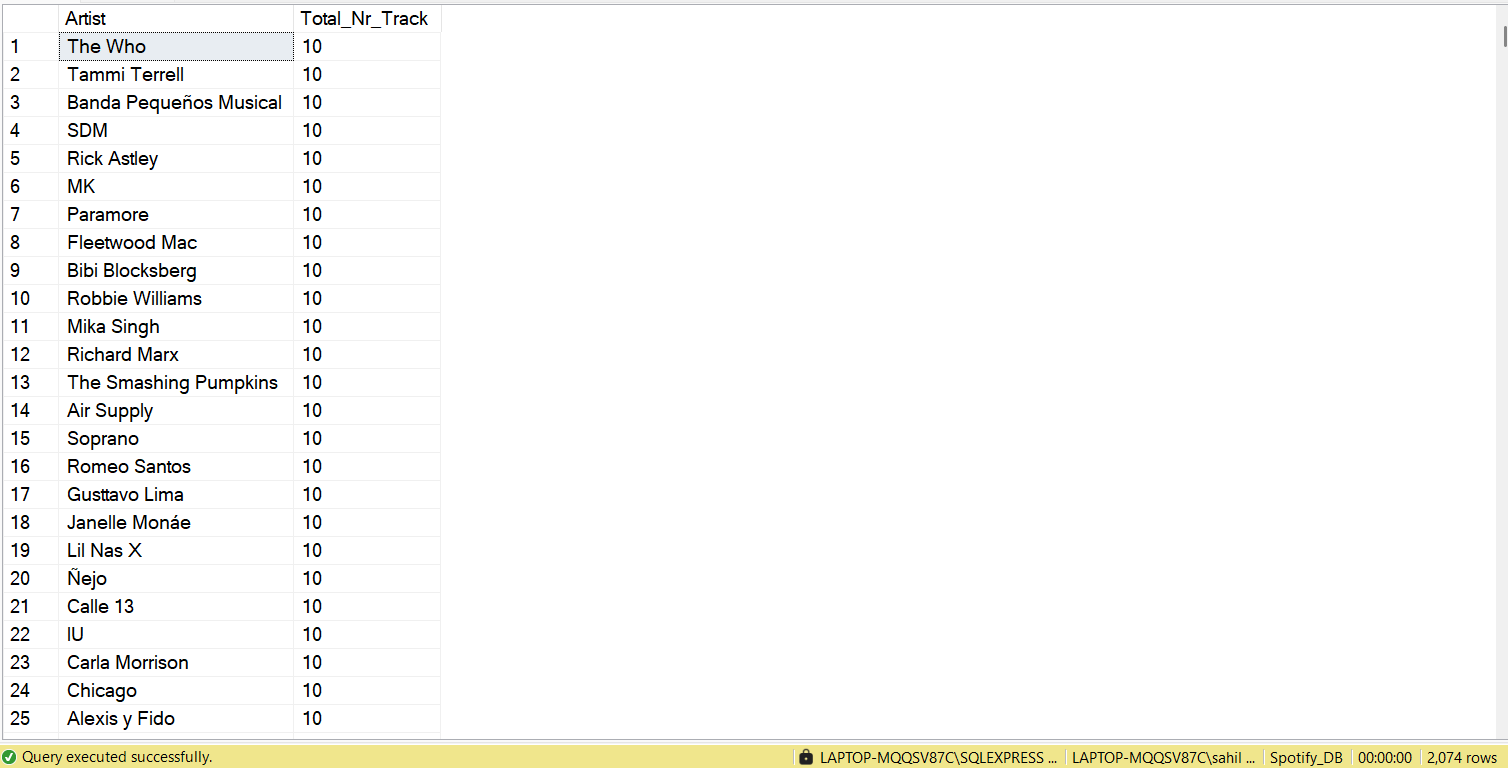
COUNT(Track) Total\_Nr\_Track

FROM Spotify\_Data

GROUP BY Artist

ORDER BY

COUNT(Track) DESC



===================================================

**Data Analysis Intermediate Level**

===================================================

(Q6) Calculate the Average Danceability of tracks in each album

SELECT

Album,

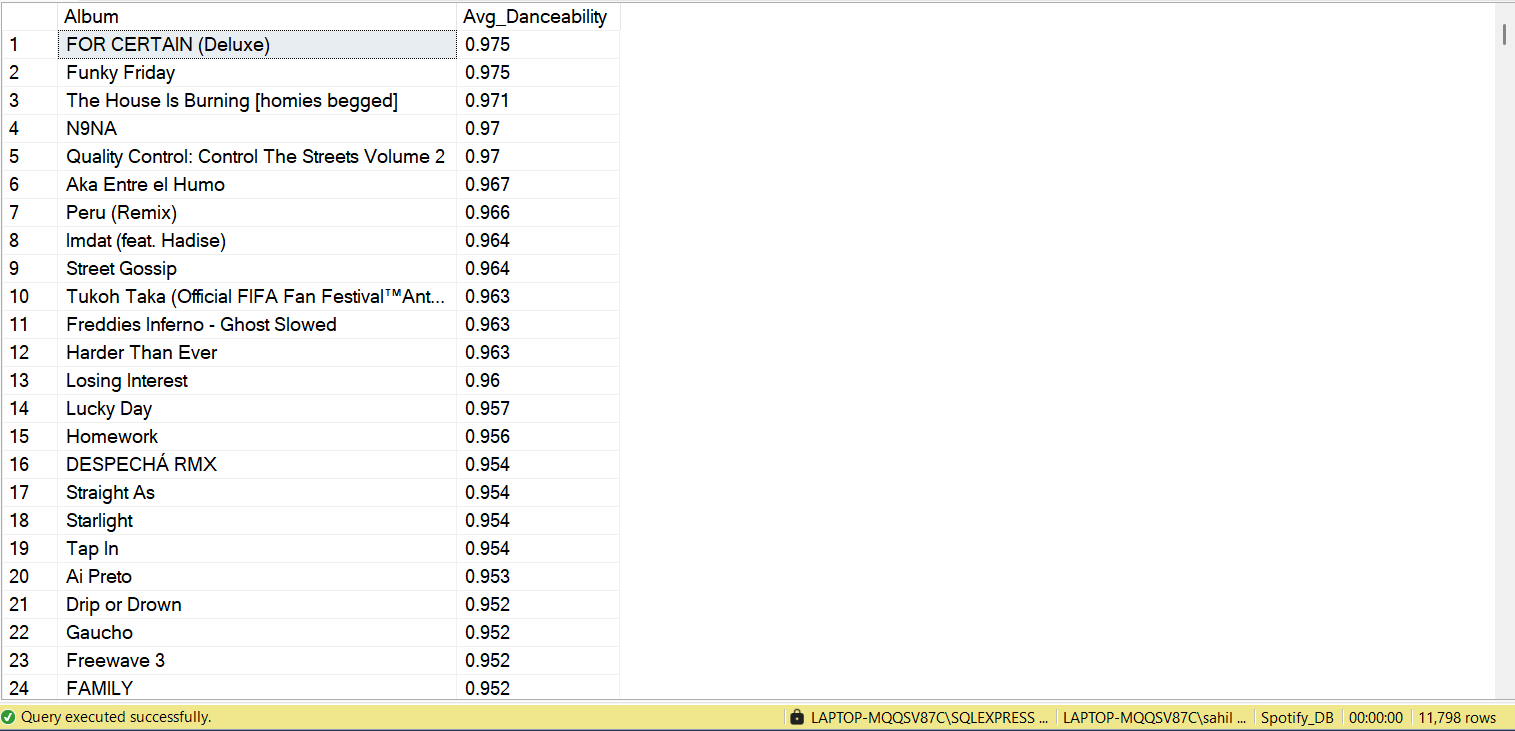
AVG(Danceability) Avg\_Danceability

FROM Spotify\_Data

GROUP BY Album

ORDER BY

AVG(Danceability) DESC



Extra thing I noted here that There are album which have more than one track

SELECT

Album,

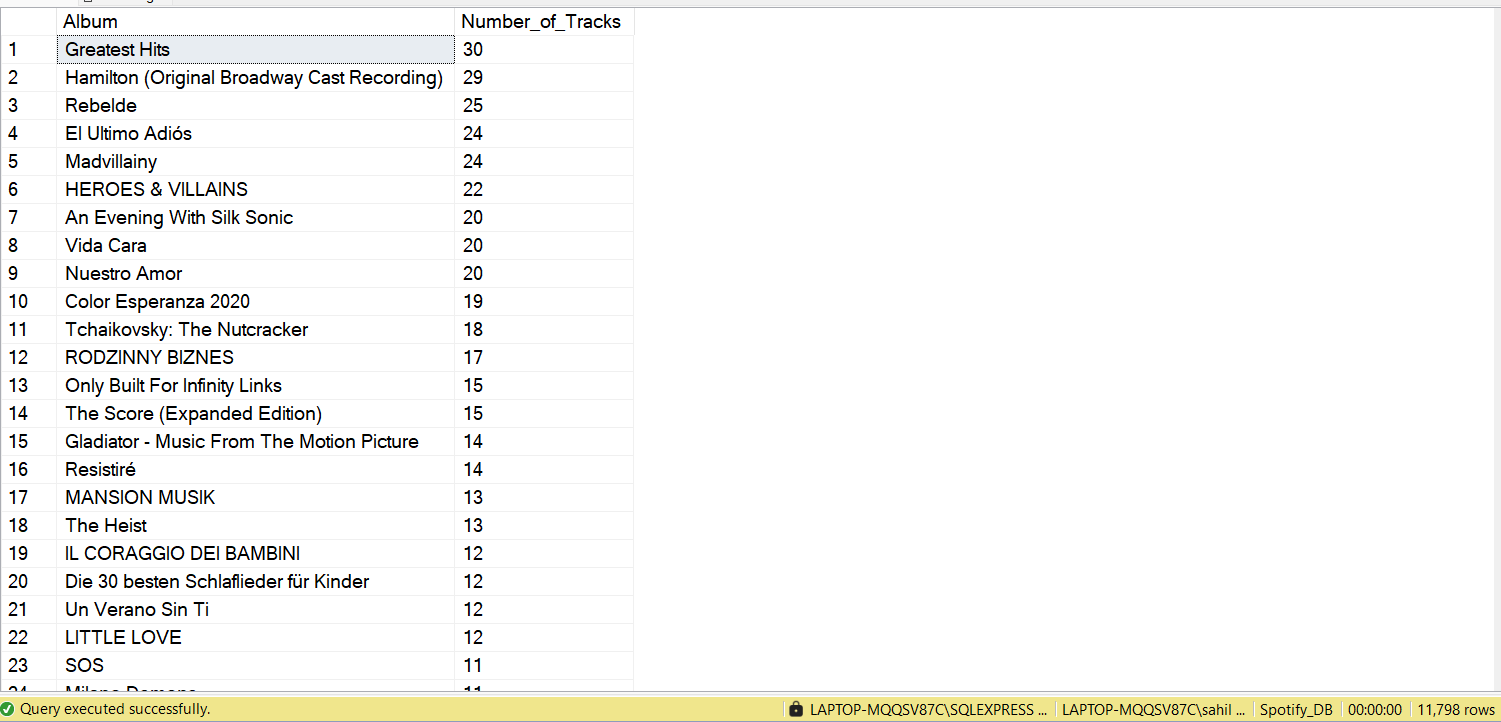
COUNT(TRACK) Number\_of\_Tracks

FROM Spotify\_Data

GROUP BY Album

ORDER BY

COUNT(TRACK) DESC



(Q7) Find the Top 5 Tracks with highest energy values

SELECT TOP 5

Track,

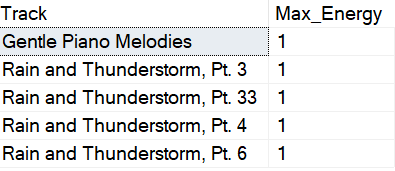
MAX(Energy) Max\_Energy

FROM Spotify\_Data

GROUP BY Track

ORDER BY

MAX(Energy) DESC



(Q8) List all tracks along with views and likes where official\_video = True

SELECT Track,

SUM(Views) Total\_Views,

SUM(Likes) Total\_Likes

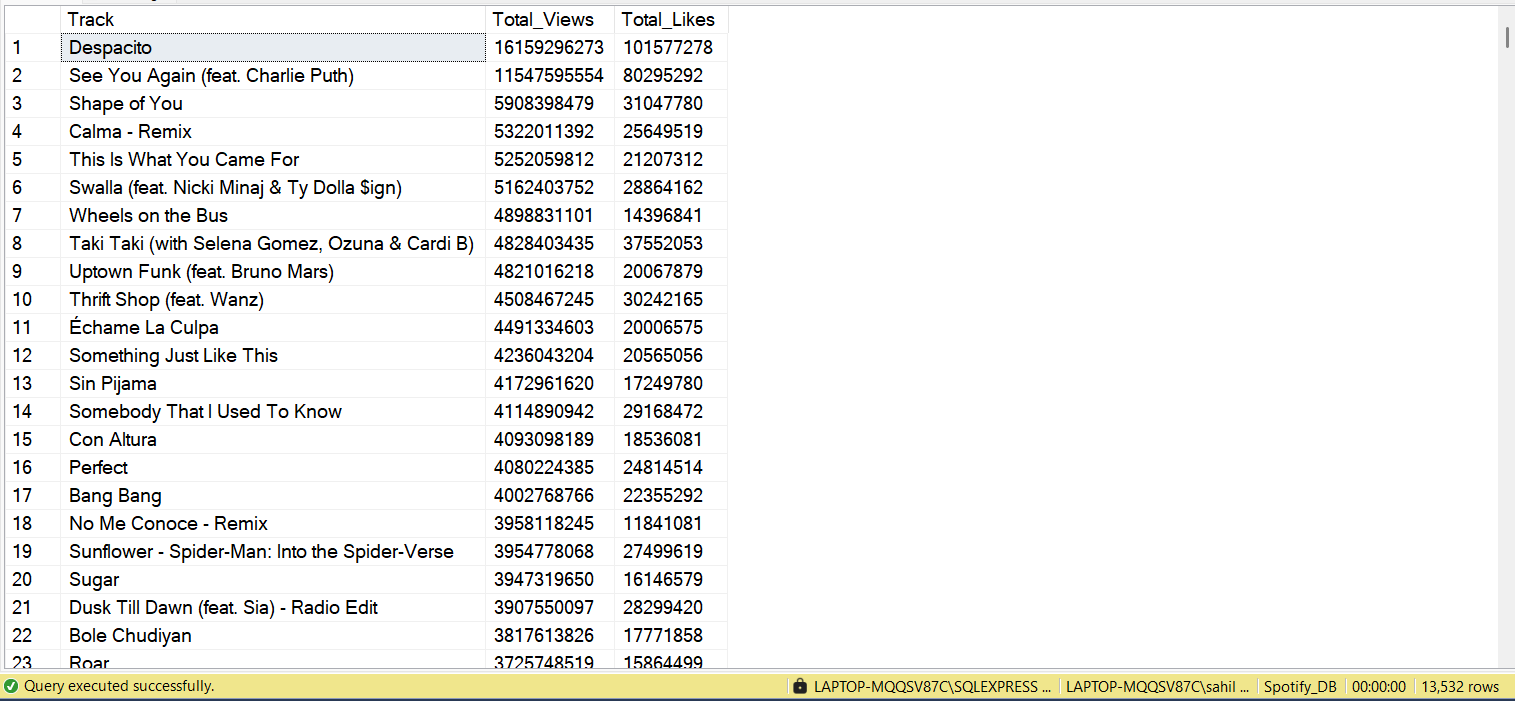
FROM Spotify\_Data

WHERE official\_video = 'TRUE'

GROUP BY Track

ORDER BY

SUM(Views) DESC



(Q9) For each album, Calculate the total views of all tracks

SELECT

ALBUM,

Track,

SUM(Views) Total\_Views

FROM Spotify\_Data

GROUP BY Track,Album

ORDER BY

SUM(Views) DESC



(Q10) Retrieve the Track names that have been streamed on Spotify more than Youtube

SELECT

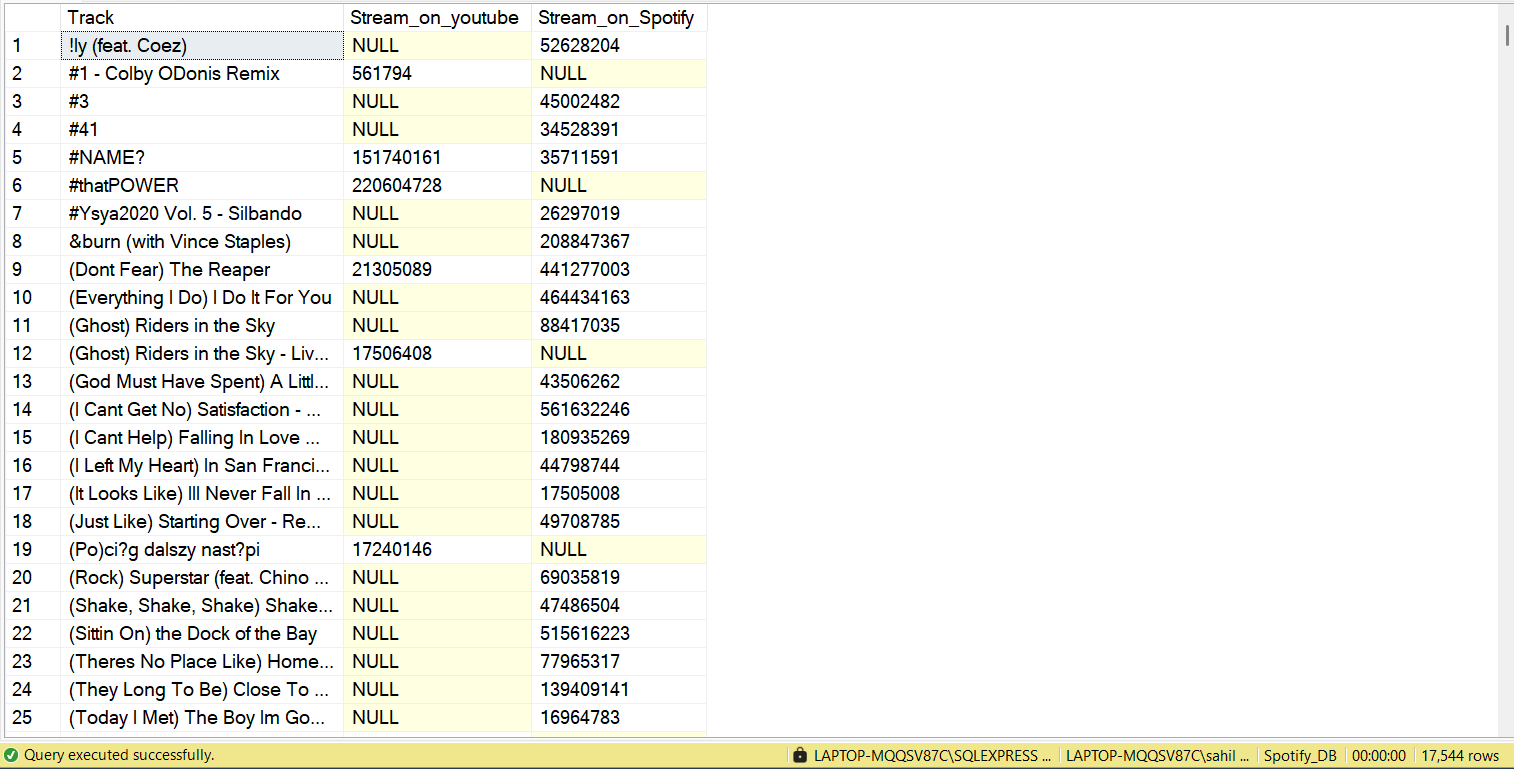
Track,

SUM(CASE WHEN most\_playedon = 'Youtube' THEN Stream END) Stream\_on\_youtube,

SUM(CASE WHEN most\_playedon = 'Spotify' THEN Stream END) Stream\_on\_Spotify

FROM Spotify\_Data

GROUP BY Track



We can Notice that Most of the songs are not streamed on both the platform

SELECT \* FROM

(SELECT

Track,

COALESCE(SUM(CASE WHEN most\_playedon = 'Youtube' THEN Stream END),0) Stream\_on\_youtube,

COALESCE(SUM(CASE WHEN most\_playedon = 'Spotify' THEN Stream END),0) Stream\_on\_Spotify

FROM Spotify\_Data

GROUP BY Track) AS T1

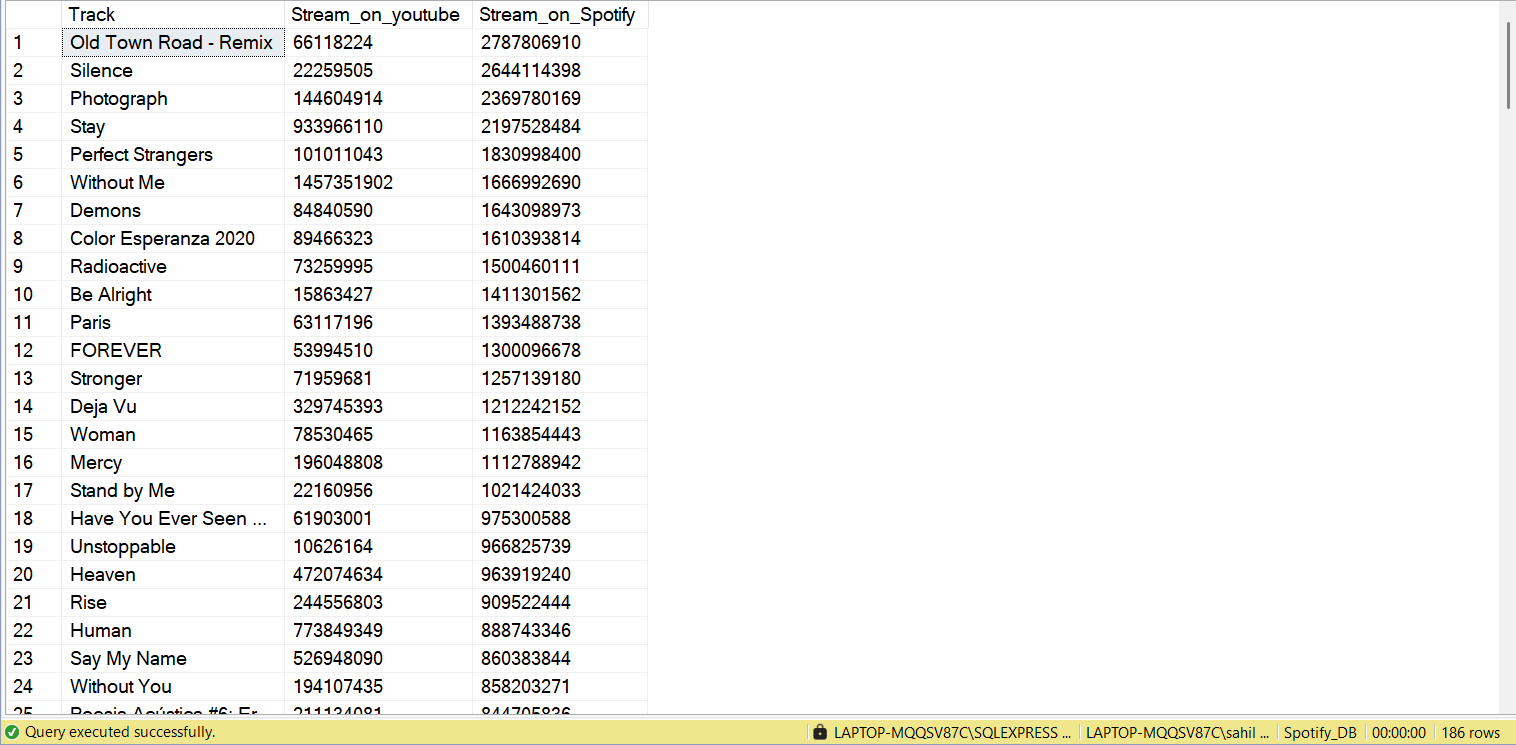
WHERE

Stream\_on\_Spotify > Stream\_on\_youtube

AND

Stream\_on\_youtube <> 0

ORDER BY Stream\_on\_Spotify DESC



===================================================

**Data Analysis Advance Level**

===================================================

(Q11) Find Top 3 most viewed tracks for each artist using window functions

-> First we will find Each Artist and Total view for each Track

-> Second Track with Highest View for each Artist( We need Top3)

-> Third use Dense Rank function and CTE

WITH Ranking\_Artist

AS(

SELECT

Artist,

Track,

SUM(Views) Total\_Views,

DENSE\_RANK() OVER(PARTITION BY Artist ORDER BY SUM(Views) DESC) Rank

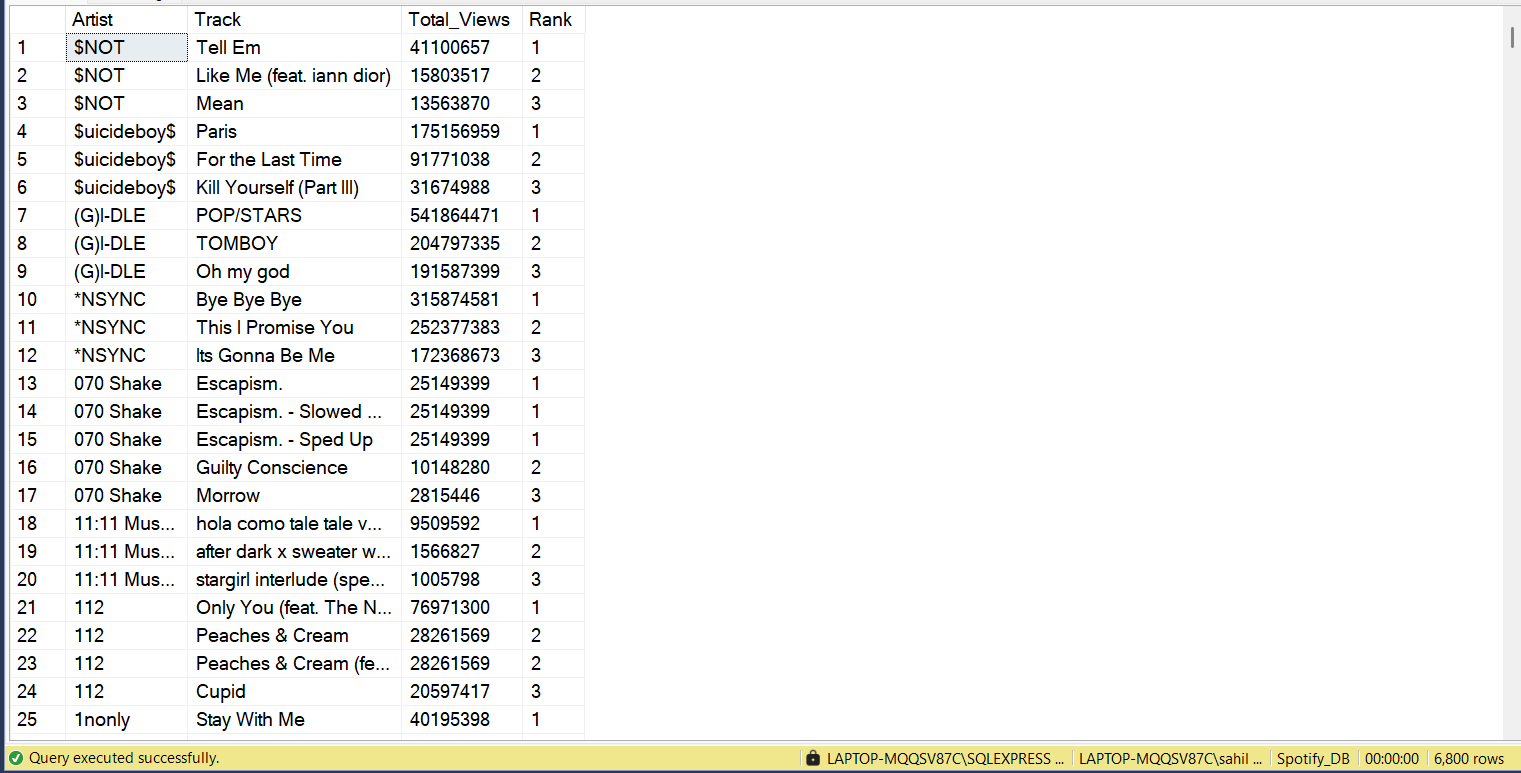
FROM Spotify\_Data

GROUP BY Artist,Track

)

SELECT \* FROM Ranking\_Artist

WHERE Rank<=3



(Q12) Write a Query to find Tracks where the liveness is above Average

SELECT

Track,

Artist,

Liveness

FROM Spotify\_Data

WHERE Liveness> (SELECT AVG(Liveness) FROM Spotify\_Data)

ORDER BY Liveness DESC



(Q13) Use a with Clause to calculate the difference between the Highest and lowest energy values for tracks in each album

WITH Energy\_of\_Album AS

(

SELECT

Album,

MAX(Energy) Highest\_Energy,

MIN(Energy) Lowest\_Energy

FROM Spotify\_Data

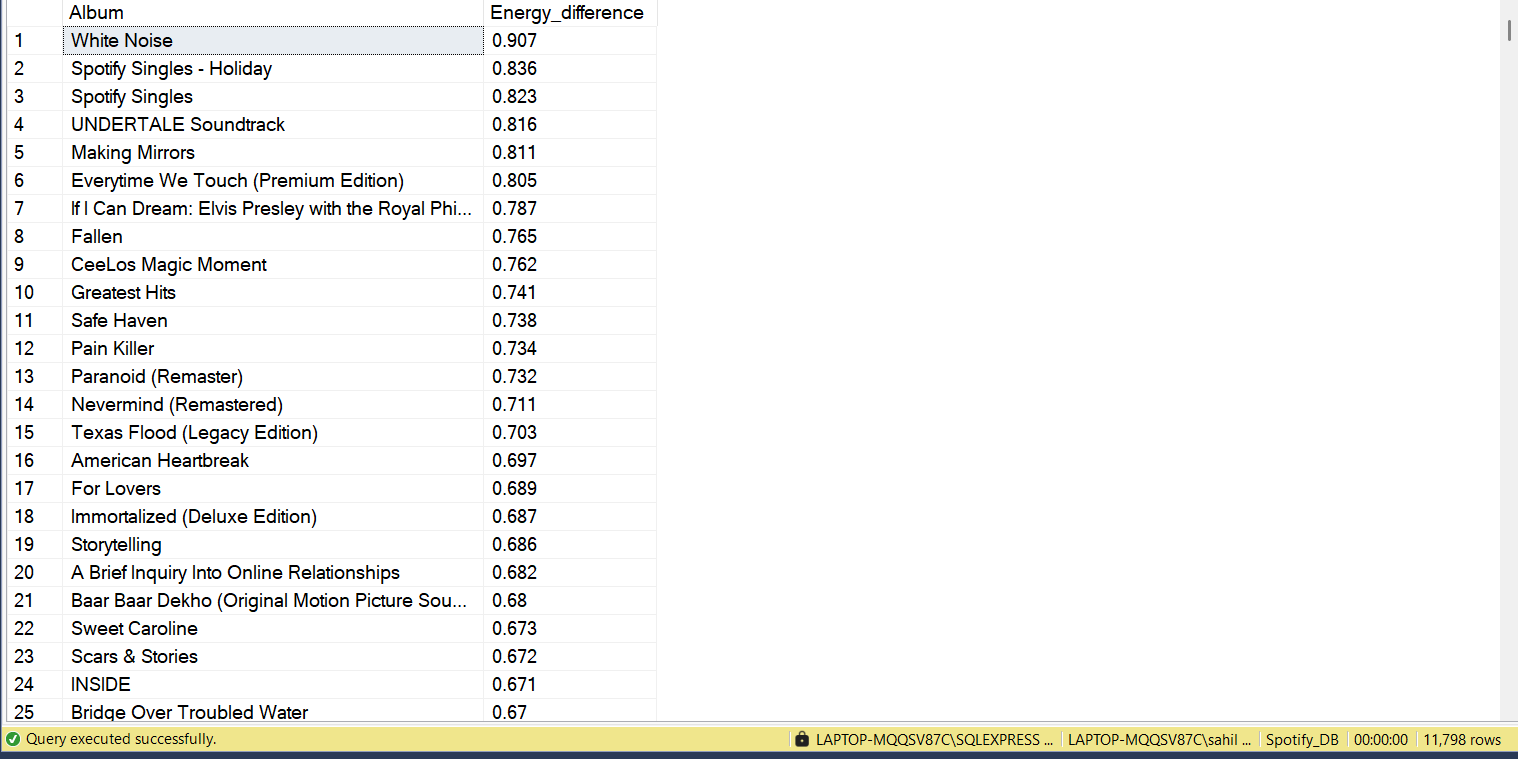
GROUP BY Album

)

SELECT Album, Highest\_Energy - Lowest\_Energy as Energy\_difference

FROM Energy\_of\_Album

ORDER BY 2 DESC



(Q14) Find tracks where the energy-to-liveness ratio is greater than 1.2.

SELECT

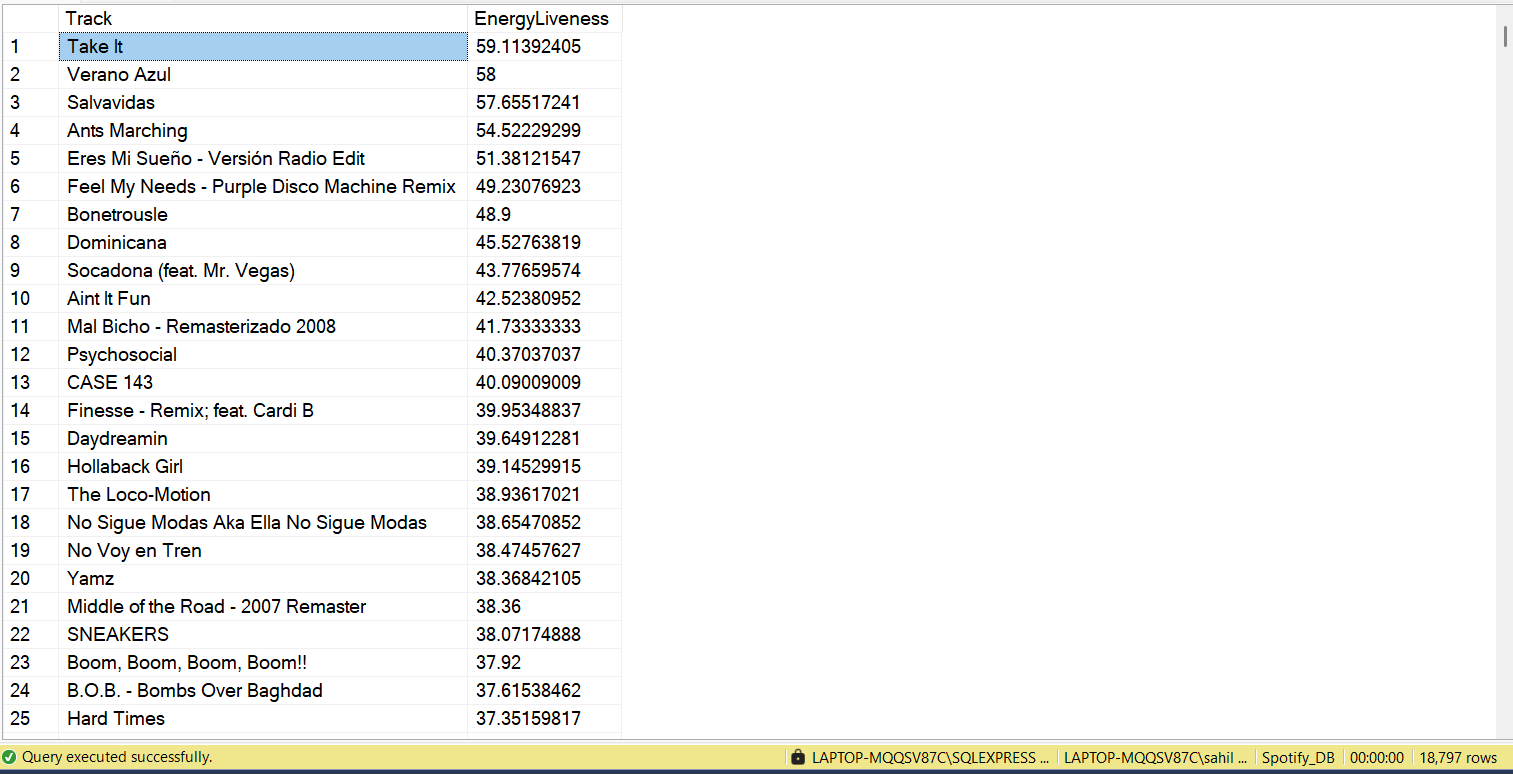
Track,

EnergyLiveness FROM Spotify\_Data

WHERE EnergyLiveness> 1.2

ORDER BY

EnergyLiveness DESC



(Q15) Calculate the cumulative sum of likes for tracks ordered by the number of views, using window functions.

SELECT

Track,

SUM(TotalLikes) OVER (ORDER BY TotalViews DESC) AS CumulativeLikes

FROM (

SELECT

Track,

SUM(Views) AS TotalViews,

SUM(likes) AS TotalLikes

FROM Spotify\_Data

GROUP BY Track

) t3

ORDER BY TotalViews DESC;



**Query Optimizations**

SELECT

TOP 25

Artist,

Track,

Views

FROM Spotify\_Data

WHERE

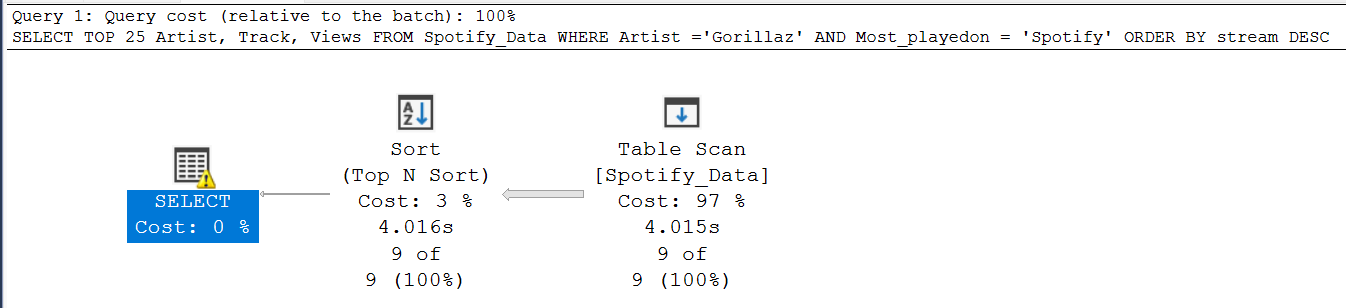
Artist ='Gorillaz'

AND

Most\_playedon = 'Spotify'

ORDER BY

stream DESC



CREATE INDEX Artist\_Index ON Spotify\_Data (Artist)

-- Ran query after creating Index

SELECT

TOP 25

Artist,

Track,

Views

FROM Spotify\_Data

WHERE

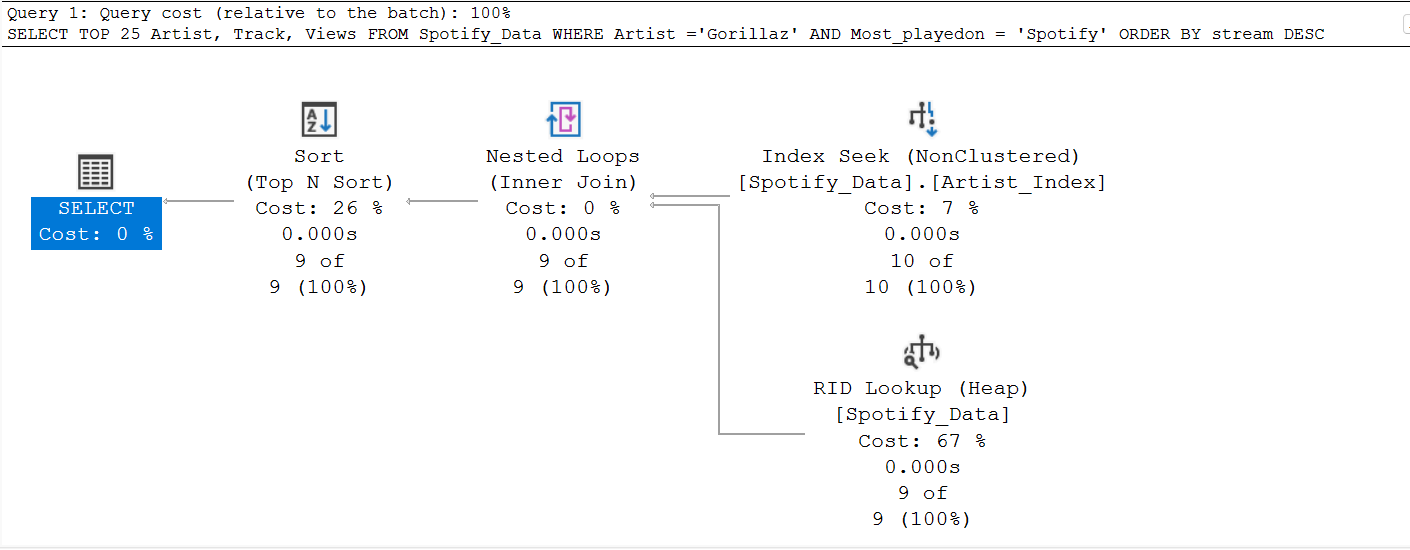
Artist ='Gorillaz'

AND

Most\_playedon = 'Spotify'

ORDER BY

stream DESC



By this way we can optimize performance of Queries