

# **Foundations of Database Management Systems**

## **BCIS 5420**

### **Fall 2021**

#### **Group Project Team 9**

#### **Analysis of Music Industry**

**Sritha Darbha**  
**Sravanthi Peri**  
**Manasa Gogineni**  
**Srinath Nallamothu**

## **CONTENTS**

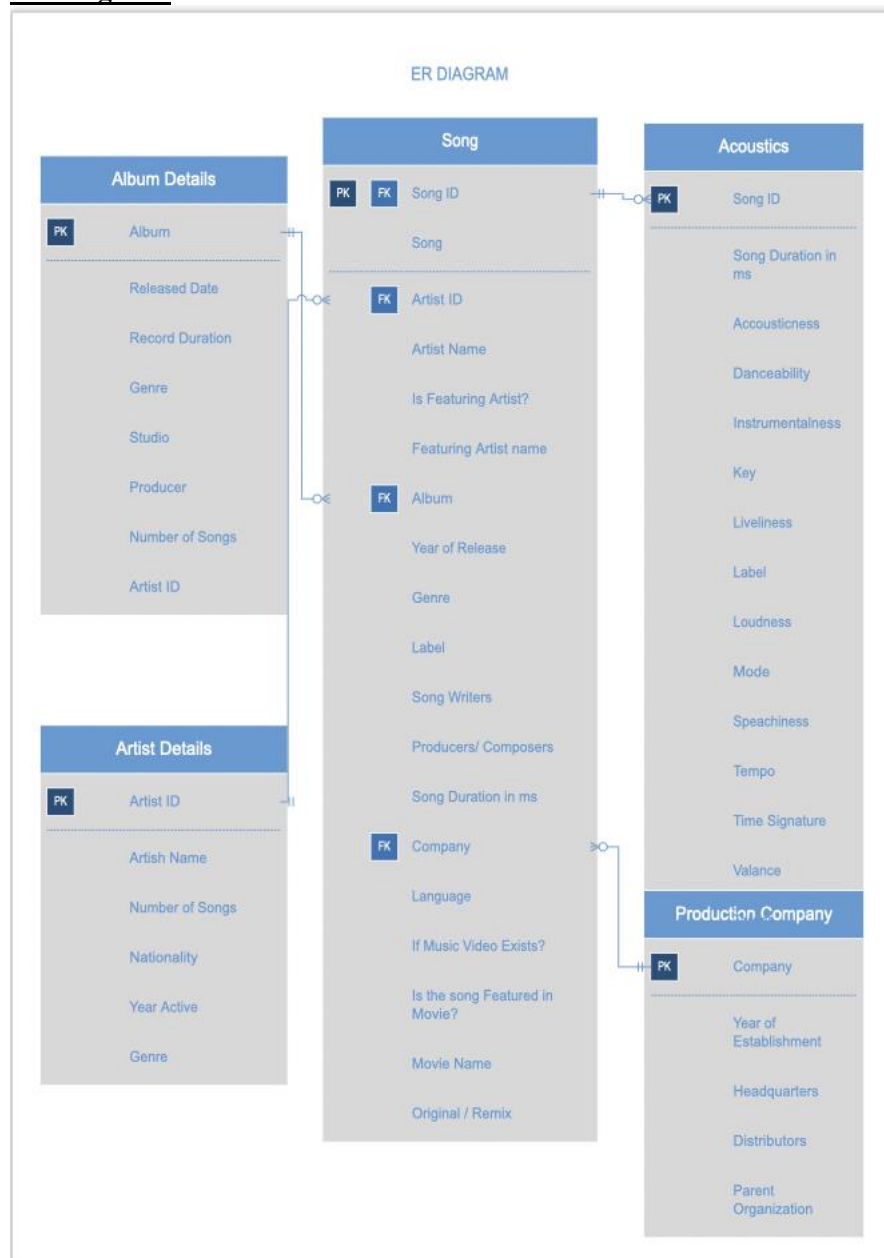
Executive Summary -----	03
Project Motivation -----	03
ER Diagram -----	04
List of Tables Created -----	05
Queries -----	07
Conclusion -----	22
References -----	22

**Executive Summary:**

This project is to analyze and extract different kinds of information related to a particular Song, Artist, Album, Production Company and much more, which usually have information over large datasets. Using MSSQL will help in reducing the time and effort in retrieving large rows of data.

**Project Motivation:**

The motivation of the project is to reduce time in retrieving large amounts of data. The dataset has five tables: with details about Songs, Acoustics, Album Details, Artist Details and Production Company. The dataset used for this report is manually written except for the table 'Acoustics', which is a second-hand dataset, taken from <https://www.kaggle.com/aeryan/spotify-music-analysis/notebook>. The dataset is information of song metadata taken from Spotify. The data contains numeric metrics generated by spotify which measure the songs' danceability, mood, liveness, etc. The data also contains the Song's title and other details, Artist Details, Album Details, Production Company details. All the Song, Artist, Album information are taken from <https://en.wikipedia.org/> and <https://www.google.co>. Usually, Record labels and Production companies have large datasets with different details about the Songs, Artists, and Albums. Using MSSQL would help in retrieving these large datasets in seconds for further analyzing the data for current Business Needs.

**ER Diagram:**

**List of Tables Created:**

We have created a total of Five tables for this Project.

- 1.Song
- 2.Acoustics
- 3.Production Company
- 4.Album Details
- 5.Artist Details

The table 'Song' has a total of 18 Entities.

Song ID
Song Name
Artist Name
Artist ID
Is Featuring Artist?
Featuring Artist Name
Album Name
Year of Release
Genre
Song Writers
Producers/ Composer
Song Duration_ms
Label
Language
If music video exists?
If the song featured in movie?
Movie name
Original/Remix

The table 'Acoustics' has a total of 15 Entities.

Song ID
Song Duration in ms
acousticness
danceability
energy
instrumentalness
key
liveness
loudness
mode
speechiness
tempo
time_signature
valence
target

The table 'Production Company' has 5 Entities.

Company
Year of establishment
Headquarters
Distributors
Parent Organization

The table 'Album Details' has 9 Entities.

Album
Released date
Record Duration - years
Genre
Studio
Label
Producer
Number of Songs
Artist ID

The table 'Artist Details' has 7 Entities.

Artist Name
Artist ID
Number of Songs
Nationality
Years Active
Label
Genre

## Queries:

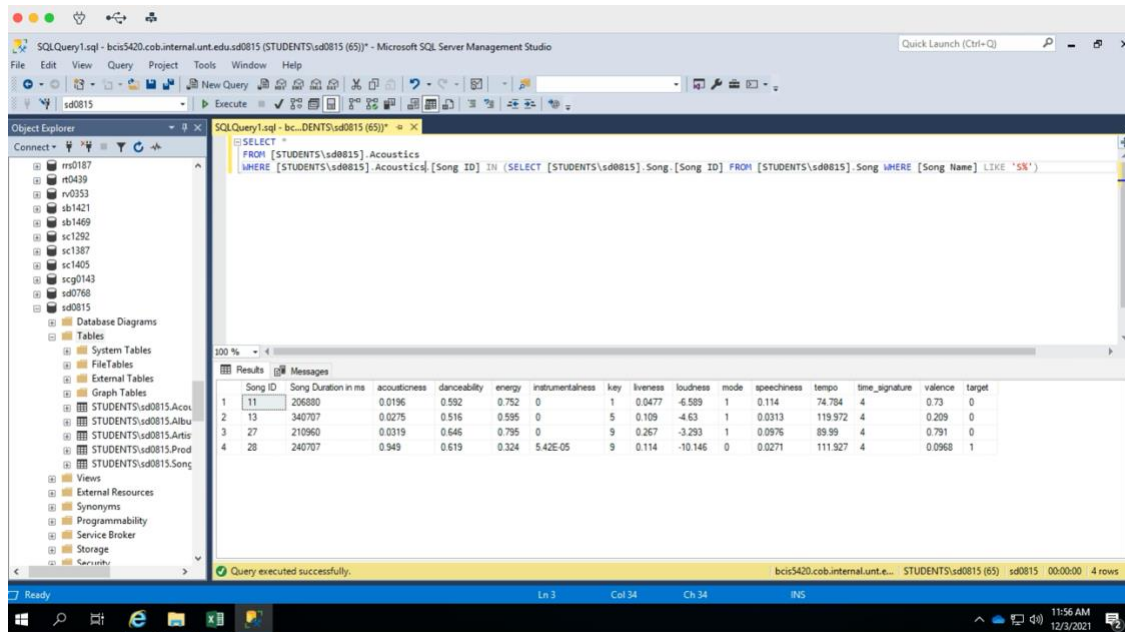
1. Give acoustic details for songs which start with S.- If someone wants to search for Songs which start with S or any alphabet.

**SELECT \***

**FROM** [STUDENTS\sd0815].Acoustics

**WHERE** [STUDENTS\sd0815].Acoustics.[Song ID] **IN** (SELECT [STUDENTS\sd0815].Song.[Song ID]

**FROM** [STUDENTS\sd0815].Song **WHERE** [Song Name] **LIKE** 'S%')



In this Query, we need to get all the acoustic details for that we are selecting all the columns from the table Acoustics and we are using the condition WHERE to select songs that start with S while using the LIKE operator.

2. List the Song details of Artists whose Nationality is American. Order the result set as per the recent releases. – If someone wants details which are created by Americans.

**SELECT \***

**FROM** [STUDENTS\sd0815].Song

**WHERE** [STUDENTS\sd0815].Song.[Artist ID] **IN** (SELECT [STUDENTS\sd0815].[Artist Details].[Artist ID ]

**FROM** [STUDENTS\sd0815].[Artist Details] **WHERE** [STUDENTS\sd0815].[Artist Details].Nationality **LIKE** 'AMERICAN')

**ORDER BY** [STUDENTS\sd0815].Song.[Year of Release] **DESC**

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```

SELECT *
FROM [STUDENTS\sd0815].Song
WHERE [STUDENTS\sd0815].Song.[Artist ID] IN (SELECT [STUDENTS\sd0815].[Artist Details].[Artist ID] FROM [STUDENTS\sd0815].[Artist Details]
WHERE [STUDENTS\sd0815].[Artist Details].[Nationality] LIKE 'AMERICAN')
ORDER BY [STUDENTS\sd0815].Song.[Year of Release] DESC

```

The Results pane displays the following data:

Song ID	Song Name	Artist Name	Artist ID	Is Featuring Artist?	Featuring Artist Name	Album Name	Year of Release	Genre	Son
1	HUMBLE.	Kendrick Lamar	1116	No	None	DAMN	2017	Hip Hop	Ker
2	Mask Off	Future	1111	No	None	Future	2017	Hip Hop	Naj
3	Mask Off	Future	1111	No	None	Future	2017	Hip Hop	Naj
4	I'm the One	DJ Khaled	1106	Yes	Justin Bieber, Quavo, Chance the Rapper, Lil Wayne	Grateful	2017	Hip Hop	D J
5	Cold	Maroon 5	1121	Yes	Future	Red Pill Blues	2017	Electronic	Adi
6	Hey Ma (with Pitbull & J Balvin feat. Camila Cab...	Pitbull	1123	Yes	Camila Cabello	The fate of the Furious	2017	Pop	Pitb
7	Attention	Charlie Puth	1102	No	None	VoiceNotes	2017	Pop	Chu
8	No Role Modelz	J. Cole	1114	No	None	Forest Hills Drive	2015	Hip Hop	J C
9	Hotline Bling	Drake	1108	No	None	Views	2015	Pop	Aut
10	Jumpman	Drake	1108	No	None	What a time to be alive	2015	Hip Hop	Naj

The status bar at the bottom indicates: Query executed successfully. bcis5420.cob.internal.unt.edu... STUDENTS\sd0815 (65) sd0815 00:00:00 21 rows

In this Query we need to list the song details of all the artists whose Nationality is American. For this we first use SELECT to select all the columns from the table [STUDENTS\sd0815].song and using a Where condition to select all the artists that are in the table [STUDENTS\sd0815].[Artist Details] and [STUDENTS\sd0815].song using their [Artist ID] and then selecting only the artists with their nationality American using the LIKE operator.

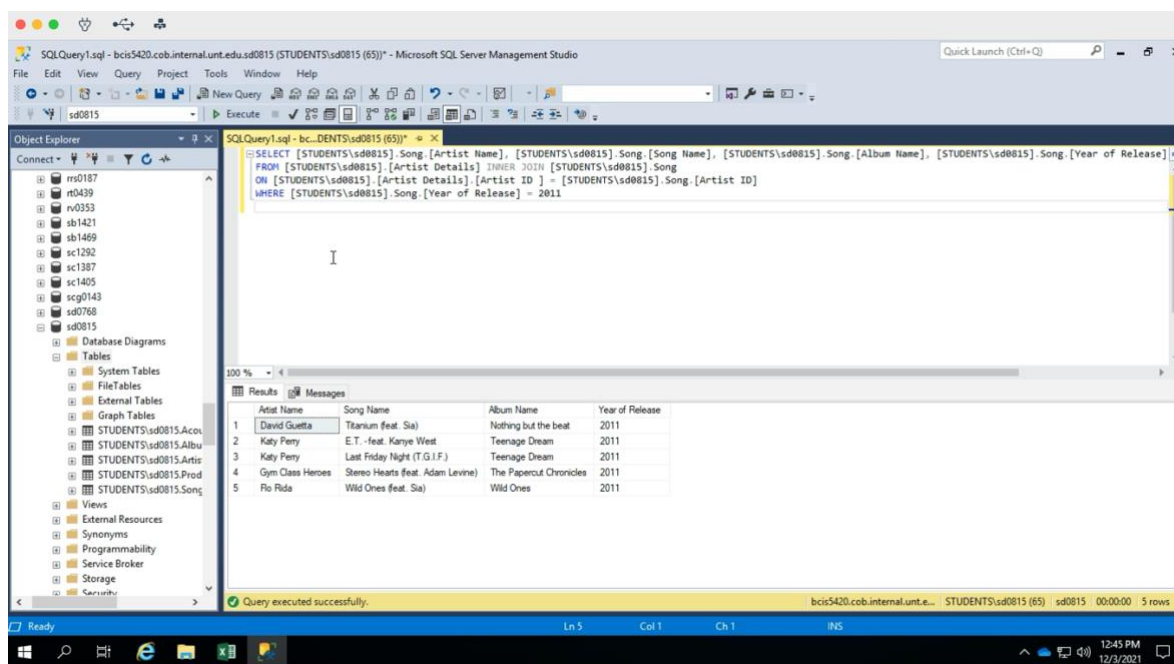
3. Show Artist Name, Song Name, Album Name, Year of Release for songs released in the year 2011. – If someone wants specific song details regarding songs which are released in 2011.

```

SELECT [STUDENTS\sd0815].Song.[Artist Name], [STUDENTS\sd0815].Song.[Song Name],
[STUDENTS\sd0815].Song.[Album Name], [STUDENTS\sd0815].Song.[Year of Release]
FROM [STUDENTS\sd0815].[Artist Details]
INNER JOIN [STUDENTS\sd0815].Song
ON [STUDENTS\sd0815].[Artist Details].[Artist ID] = [STUDENTS\sd0815].Song.[Artist ID]
WHERE [STUDENTS\sd0815].Song.[Year of Release] = 2011

```

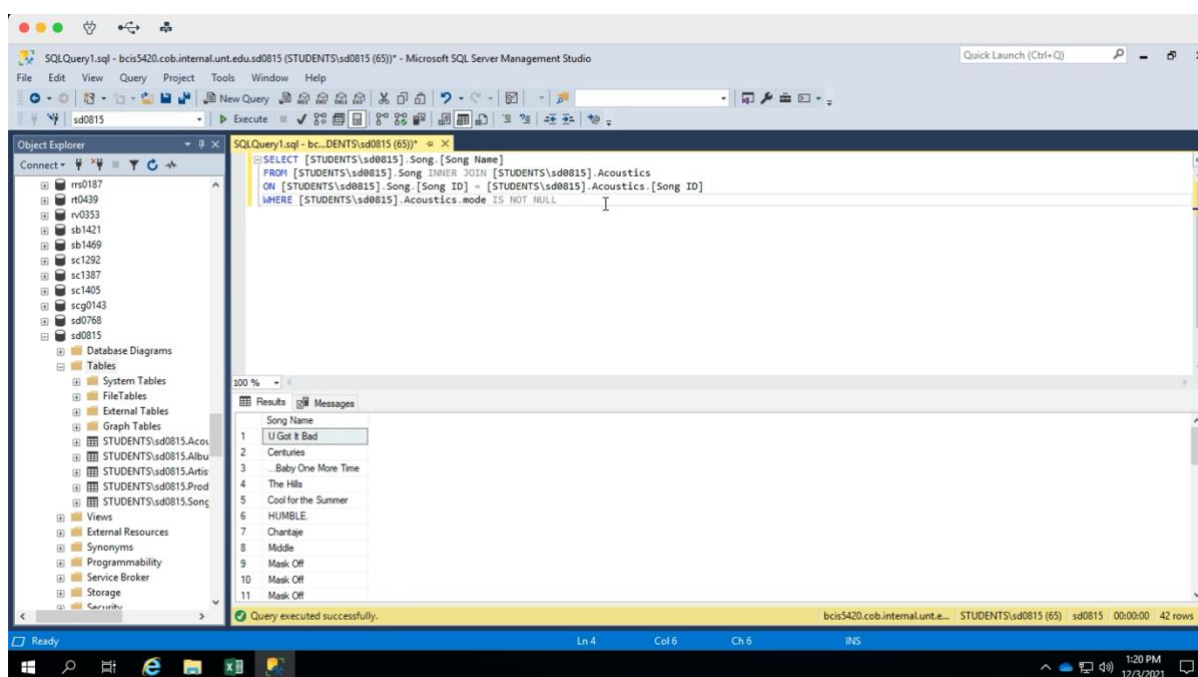




In this Query we are trying to show Artist Name, Song Name, Album Name, Year of Release for songs released in the year 2011 for this we use the SELECT statement and specify the columns required and then use INNER JOIN to join the tables Artist Details and Song using the Artist ID and use the WHERE condition to specify the year of release=2011.

4. Show the list of songs with mode not null. – If someone wants songs with mode as not null.

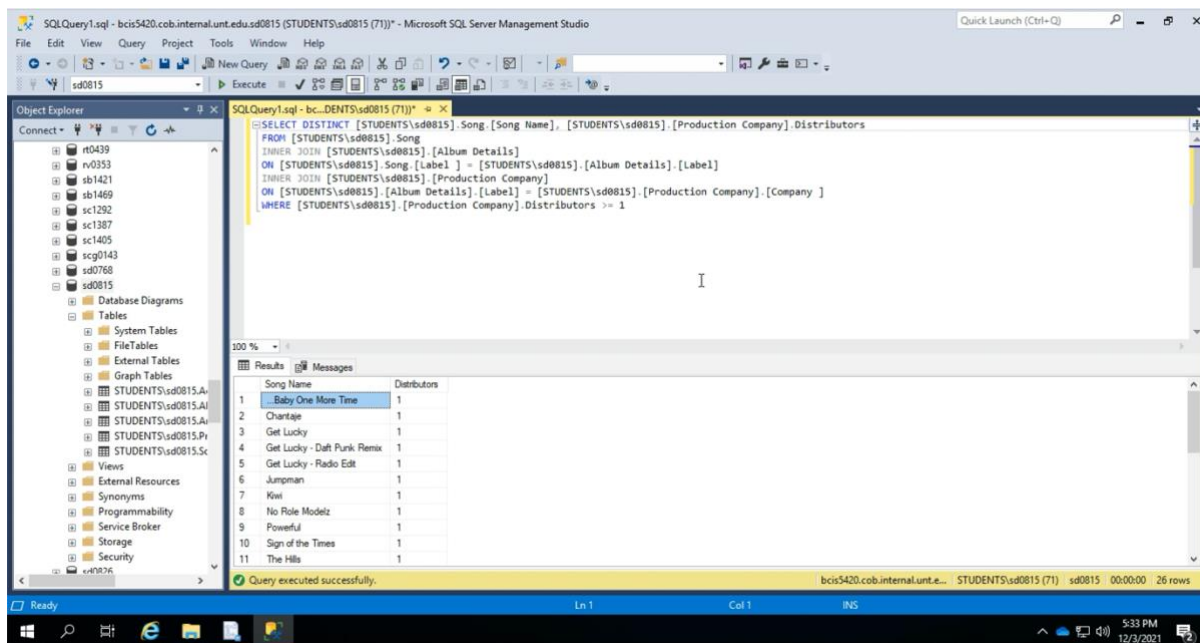
```
SELECT [STUDENTS\sd0815].Song.[Song Name]
FROM [STUDENTS\sd0815].Song INNER JOIN [STUDENTS\sd0815].Acoustics
ON [STUDENTS\sd0815].Song.[Song ID] = [STUDENTS\sd0815].Acoustics.[Song ID]
WHERE [STUDENTS\sd0815].Acoustics.mode IS NOT NULL
```



In this Query we need the list of songs with mode not null so for this we have to first select the songname from the table [STUDENTS\sd0815].Song and inner join with the table [STUDENTS\sd0815].Acoustics using the common column Song ID and then using the condition [STUDENTS\sd0815].Acoustics.mode IS NOT NULL

5. List of Songs whose album has distributors more than or equal to 1. – If someone wants songs list from albums from Companies which have more than 1 distributors.

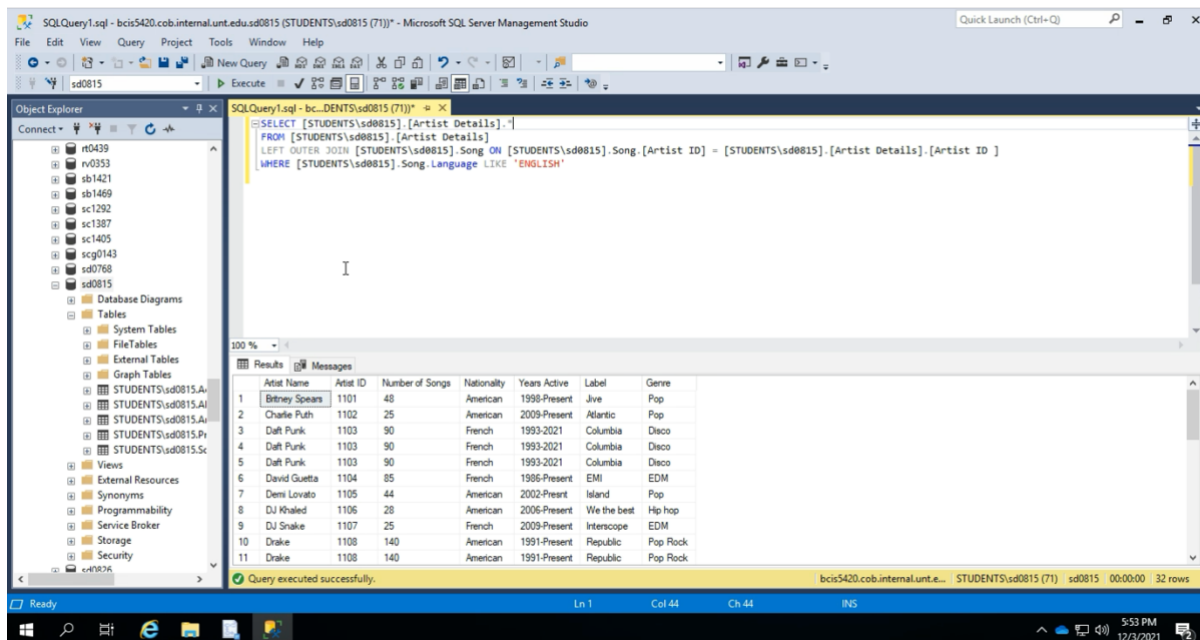
```
SELECT DISTINCT [STUDENTS\sd0815].Song.[Song Name], [STUDENTS\sd0815].[Production Company].Distributors
FROM [STUDENTS\sd0815].Song
INNER JOIN [STUDENTS\sd0815].[Album Details]
ON [STUDENTS\sd0815].Song.[Label ] = [STUDENTS\sd0815].[Album Details].[Label]
INNER JOIN [STUDENTS\sd0815].[Production Company]
ON [STUDENTS\sd0815].[Album Details].[Label] = [STUDENTS\sd0815].[Production Company].[Company ]
WHERE [STUDENTS\sd0815].[Production Company] .Distributors >= 1
```



In this Query we need to find list of songs whose album has distributors  $\geq 1$  so for this we first use select to display the column song name from the table song and column distributors from the table production company. Then using inner join, we join the tables song and album details with the common column label and then join the table production company. We use the WHERE condition to choose companies with distributors  $\geq 1$ .

6. Find all details of the artist whose language is English. – If someone wants songs based on Language.

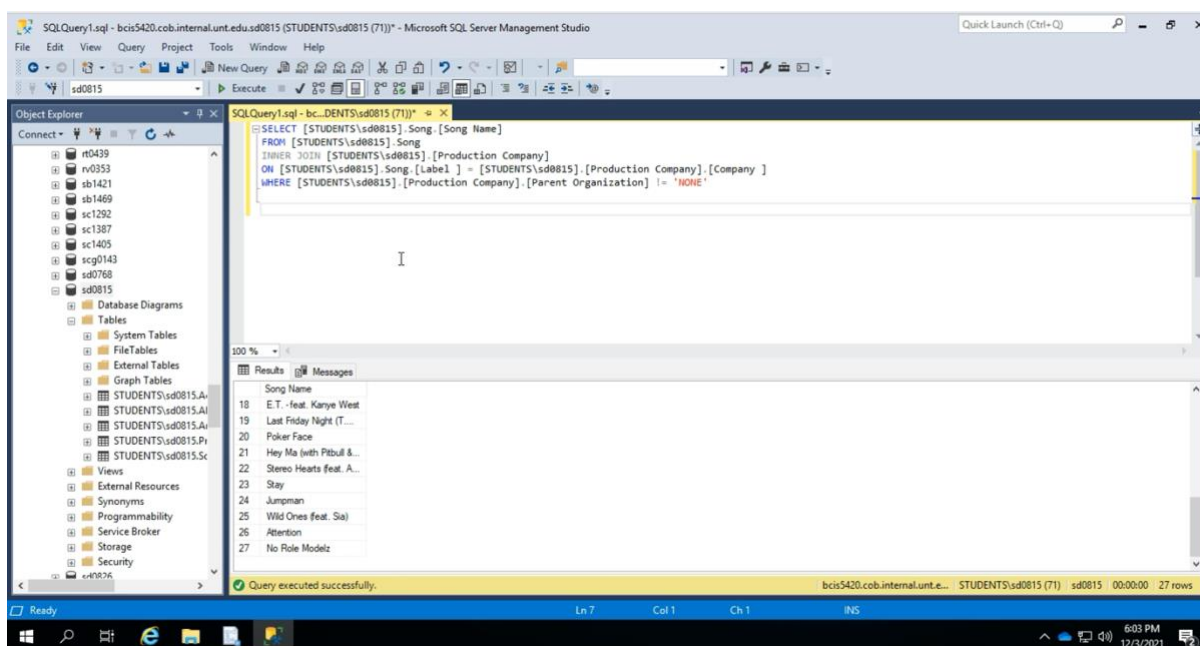
```
SELECT [STUDENTS\sd0815].[Artist Details].*
FROM [STUDENTS\sd0815].[Artist Details]
LEFT OUTER JOIN [STUDENTS\sd0815].Song ON [STUDENTS\sd0815].Song.[Artist ID] =
[STUDENTS\sd0815].[Artist Details].[Artist ID ]
WHERE [STUDENTS\sd0815].Song.Language LIKE 'ENGLISH'
```



In order to display all the details of the artist whose language is English we first use select statement in which we use \* to select all the columns from the table artist details and then use left outer join to join the tables artist details and song and where condition to select language as English.

7. Find the list of songs whose parent organization is not none. – If someone wants songs based on a characteristic of Production Company.

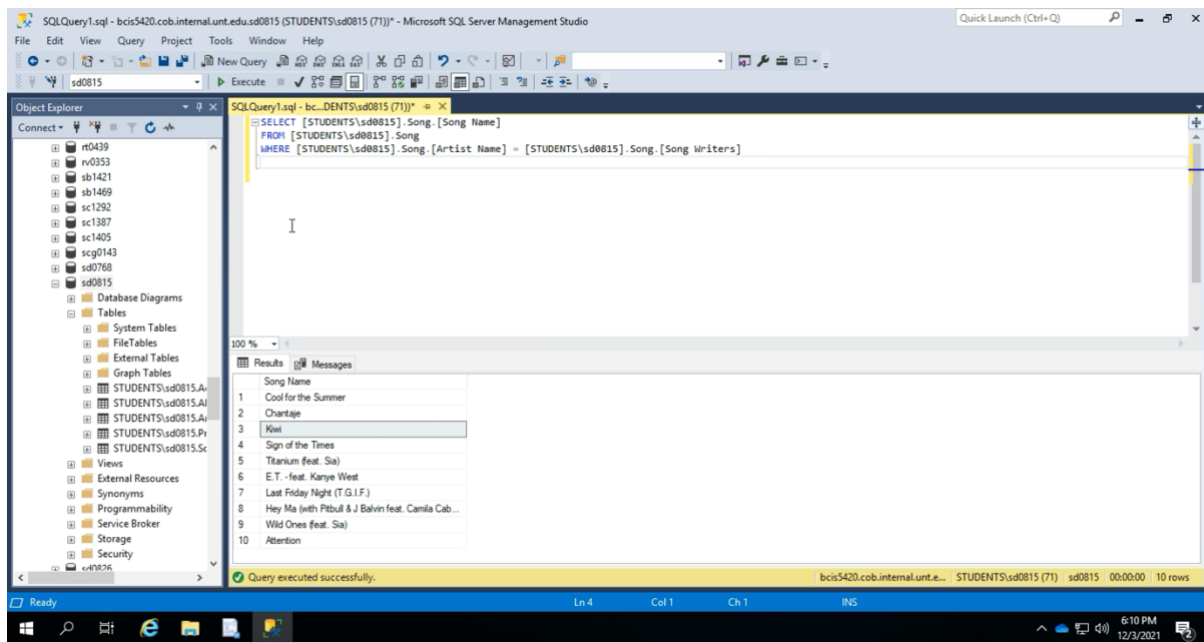
```
SELECT [STUDENTS\sd0815].[Song].[Song Name]
FROM [STUDENTS\sd0815].[Song]
INNER JOIN [STUDENTS\sd0815].[Production Company]
ON [STUDENTS\sd0815].[Song].[Label] = [STUDENTS\sd0815].[Production Company].[Company]
WHERE [STUDENTS\sd0815].[Production Company].[Parent Organization] != 'None'
```



In this Query we use the select statement to select the column song name from the table song and use inner join to join the tables song and production company using the common column company and then specifying the condition where parent organization != none.

8. List out the songs which were performed and written by the same Artist. – If someone wants details of Songs which were performed and written by same artist.

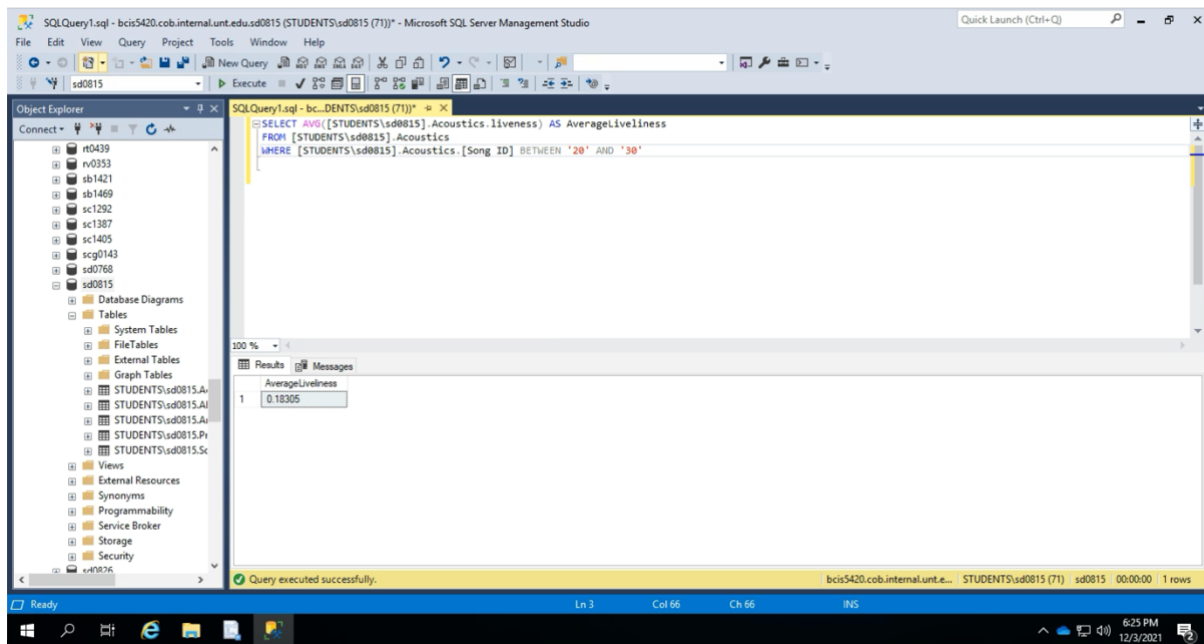
```
SELECT [STUDENTS\sd0815].Song.[Song Name]
FROM [STUDENTS\sd0815].Song
WHERE [STUDENTS\sd0815].Song.[Artist Name] = [STUDENTS\sd0815].Song.[Song Writers]
```



In this query we are trying to list out the songs which were performed and written by the same artist so first we use select statement to select the column song name from the table [[STUDENTS\sd0815].Song and then using the where condition to choose the songs that hav same artist and writers.

9. Display Average Liveliness for Songs with ID's between 20 and 30. -If someone wants Acoustic details of specified songs.

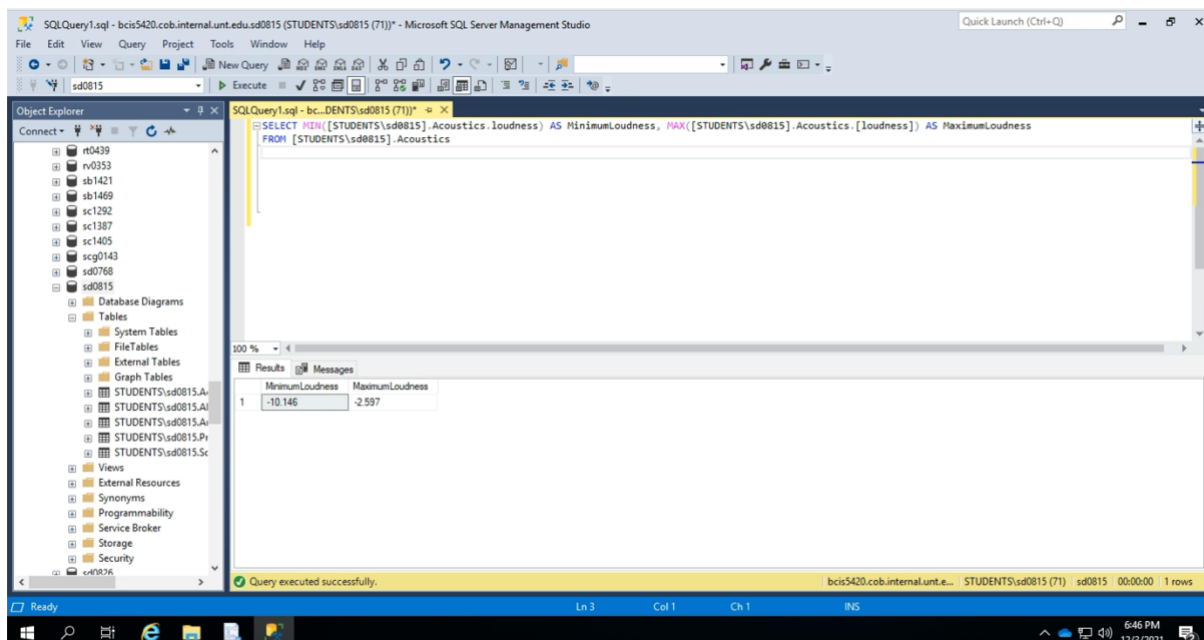
```
SELECT AVG([STUDENTS\sd0815].Acoustics.liveness) AS AverageLiveliness
FROM [STUDENTS\sd0815].Acoustics
WHERE [STUDENTS\sd0815].Acoustics.[Song ID] BETWEEN '20' AND '30'.
```



In this, we select the column Liveliness and perform the math function average on it and rename it as AverageLiveliness from the table Acoustics and use the where condition and the Between operator to choose song id's between 20-30.

10. List the Minimum and Maximum Loudness of all songs. – If someone wants songs based on song Acoustics.

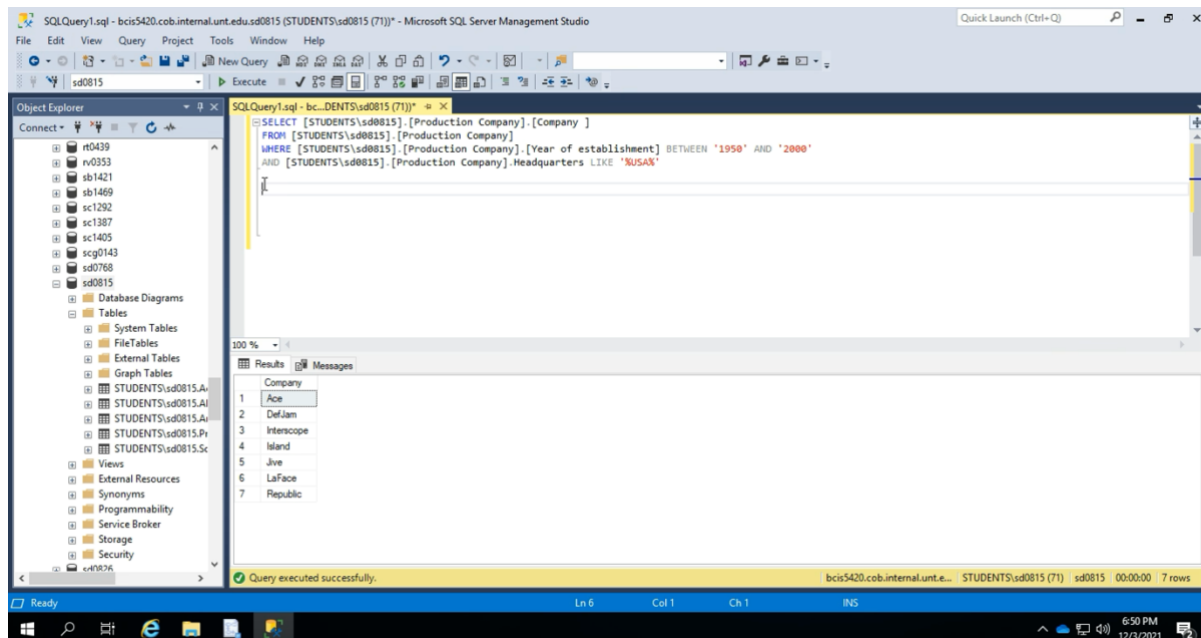
```
SELECT MIN([STUDENTS\sd0815].Acoustics.loudness) AS MinimumLoudness,
MAX([STUDENTS\sd0815].Acoustics.[loudness]) AS MaximumLoudness
FROM [STUDENTS\sd0815].Acoustics
```



In this Query, in order to display the maximum and minimum loudness we first use the select statement and select the column loudness and use the math function min and max on it and rename them as MinimumLoudness and MaximumLoudness respectively from the table Acoustics.

11. List of companies who are established in the latter half of 20th century and whose headquarters are in USA. – If someone wants songs based on the Company details.

```
SELECT [STUDENTS\sd0815].[Production Company].[Company]
FROM [STUDENTS\sd0815].[Production Company]
WHERE [STUDENTS\sd0815].[Production Company].[Year of establishment] BETWEEN '1950' AND '2000'
AND [STUDENTS\sd0815].[Production Company].Headquarters LIKE '%USA%'
```



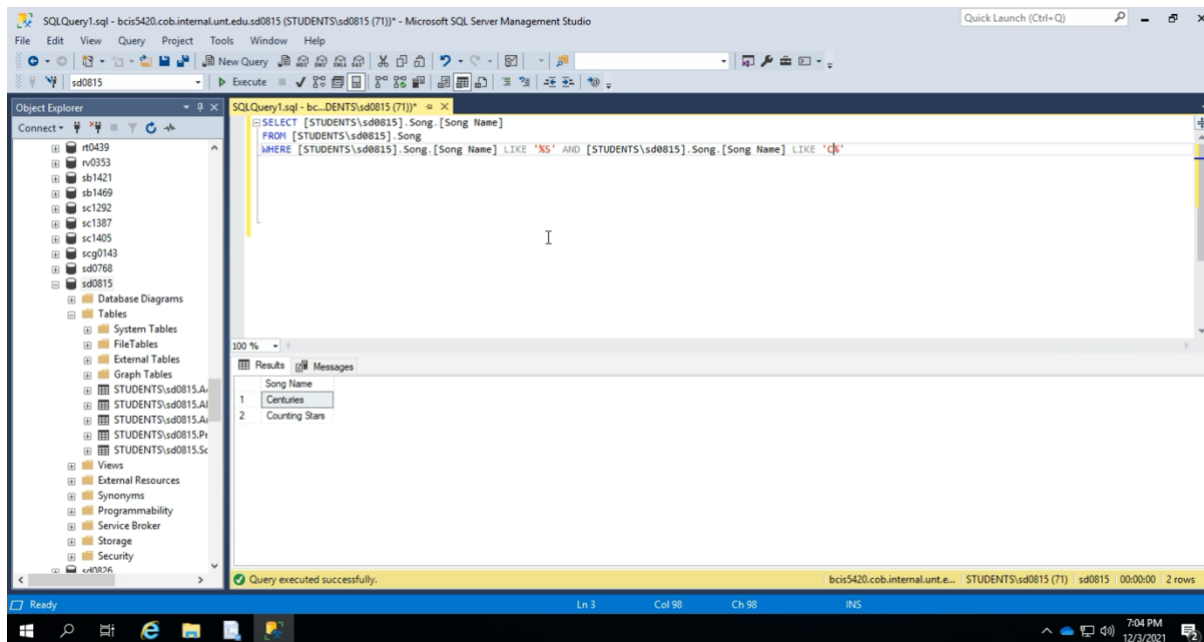
We use the select statement to select the company column from the table production company and then use where condition to specify both the conditions that year of establishment should be between 1950-2000 and the headquarters should be in USA.

For the condition year of establishment, we use the between operator and for the condition headquarters we use the operator like.

12.To List the Song which start with C and ends with S. – If someone wants songs based on song name.

```
SELECT [STUDENTS\sd0815].Song.[Song Name]
FROM [STUDENTS\sd0815].Song
WHERE [STUDENTS\sd0815].Song.[Song Name] LIKE '%S' AND [STUDENTS\sd0815].Song.[Song Name]
LIKE 'C%'
```

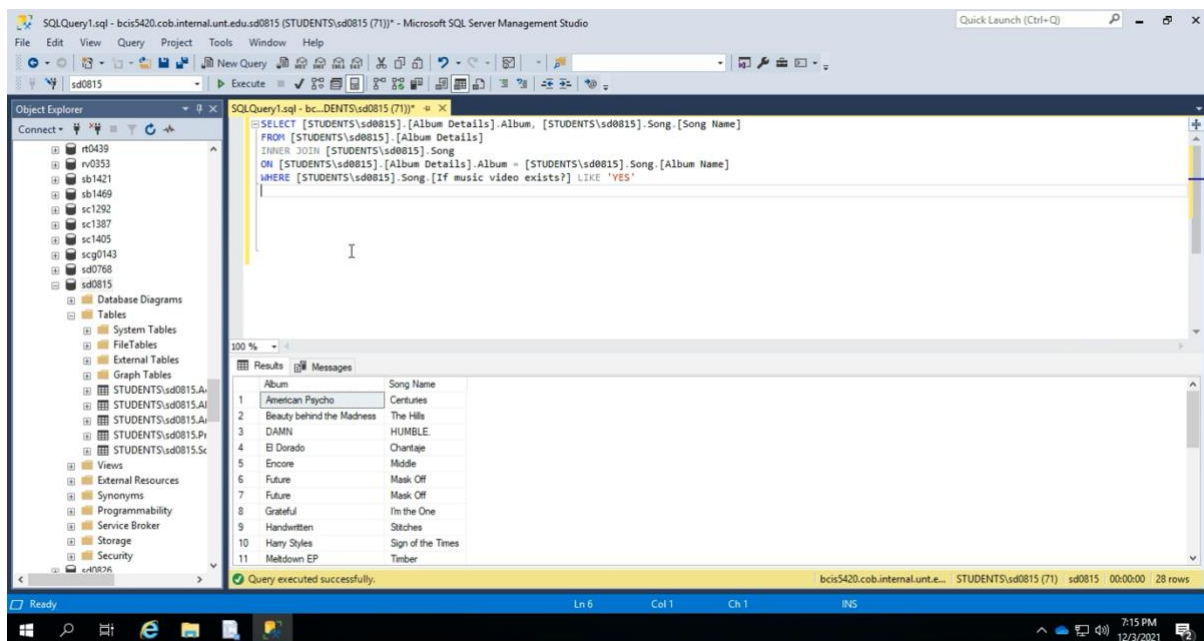




In order to display the songs whose first letter is C and last letter is S we first select the song name from the table song where we use to like operators to get the conditions. In the first condition we use 'C%' to get the song names starting with C and then use the condition '%S' to get the names of the songs ending with S.

13. Display Album names which has atleast one Video song and display the song name – If someone wants songs which have music videos.

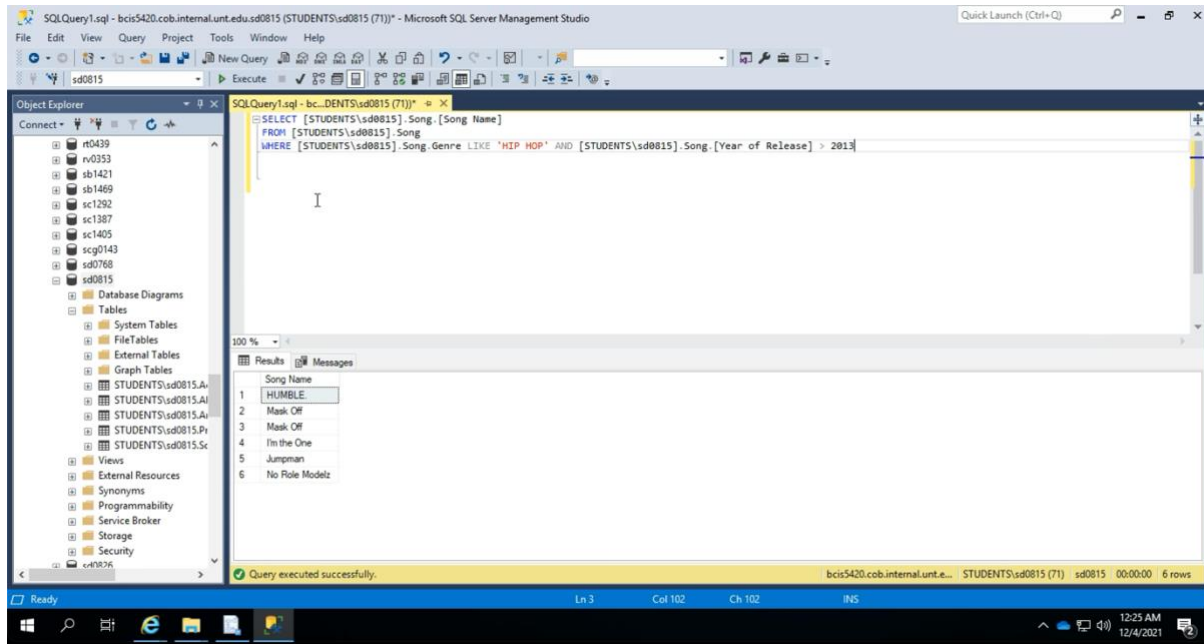
```
SELECT [STUDENTS\sd0815].[Album Details].Album, [STUDENTS\sd0815].Song.[Song Name]
FROM [STUDENTS\sd0815].[Album Details]
INNER JOIN [STUDENTS\sd0815].Song
ON [STUDENTS\sd0815].[Album Details].Album = [STUDENTS\sd0815].Song.[Album Name]
WHERE [STUDENTS\sd0815].Song.[If music video exists?] LIKE 'YES'
```



For this Query we first select the columns Album and Song name from two different tables album details and song using the inner join and we use the condition if music video exists 'yes' in the Where condition.

14. Give list of Hip Hop songs released after 2013.- If someone wants songs based on Genre and year of release.

```
SELECT [STUDENTS\sd0815].Song.[Song Name]
FROM [STUDENTS\sd0815].Song
WHERE [STUDENTS\sd0815].Song.Genre LIKE 'HIP HOP' AND [STUDENTS\sd0815].Song.[Year of Release] > 2013
```

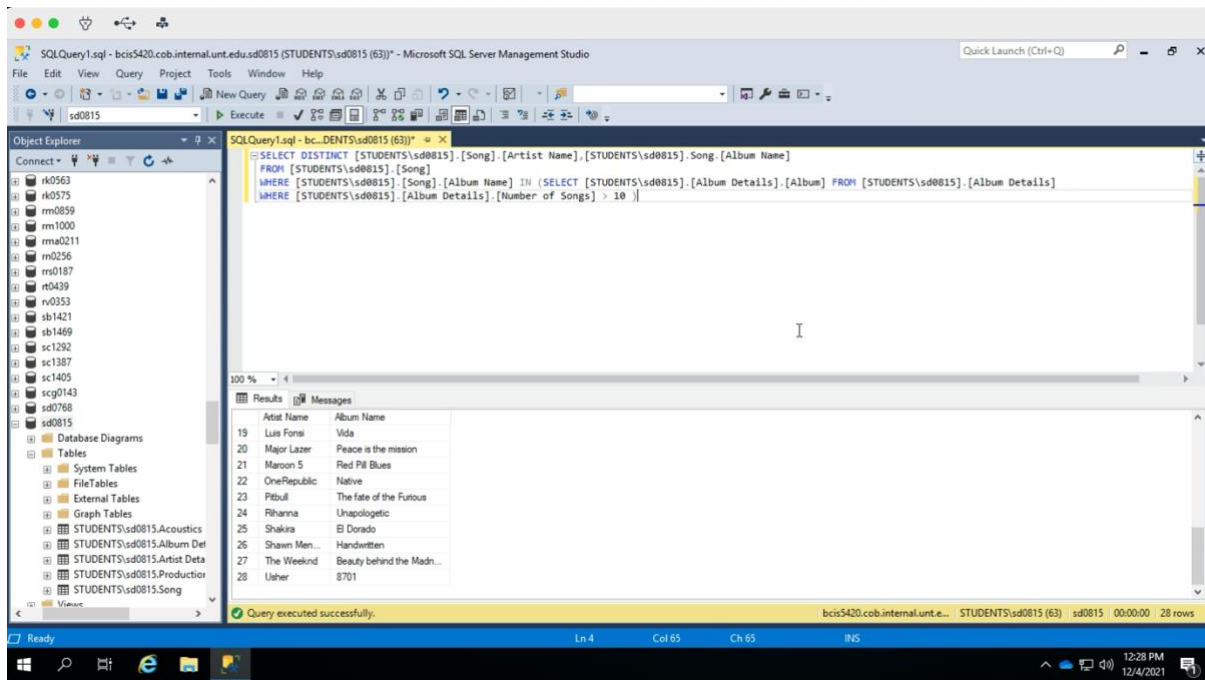


In this table we first select the song name column from the table [STUDENTS\sd0815].[Song] and in where condition we use the like condition and select hip hop as the song genre and select songs that released after the year 2013 with > symbol.

15. Give Artist name and album name for albums with songs count greater than 10. – If someone wants songs based on Song count in Album.

```
SELECT DISTINCT [STUDENTS\sd0815].[Song].[Artist Name], [STUDENTS\sd0815].Song.[Album Name]
FROM [STUDENTS\sd0815].[Song]
WHERE [STUDENTS\sd0815].[Song].[Album Name] IN (SELECT [STUDENTS\sd0815].[Album Details].[Album] FROM [STUDENTS\sd0815].[Album Details] WHERE [STUDENTS\sd0815].[Album Details].[Number of Songs] > 10)
```





In this query we select the columns artist name and album name from the table song and using a sub query to select albums which have song count greater than 10.

16. List out the Song Name, Song Duration in ms, Album, Artist Name, Company by joining all tables.- If someone wants song details like song length, album, artist and company name.

```
SELECT DISTINCT [STUDENTS\sd0815].[Song].[Song Name], [STUDENTS\sd0815].[Acoustics].[Song Duration
in ms], [STUDENTS\sd0815].[Album Details].[Album], [STUDENTS\sd0815].[Artist Details].[Artist Name],
[STUDENTS\sd0815].[Production Company].[Company ]
FROM [STUDENTS\sd0815].[Song]
INNER JOIN [STUDENTS\sd0815].[Acoustics] ON [STUDENTS\sd0815].[Song].[Song ID] =
[STUDENTS\sd0815].[Acoustics].[Song ID]
INNER JOIN [STUDENTS\sd0815].[Album Details] ON [STUDENTS\sd0815].[Song].[Artist ID] =
[STUDENTS\sd0815].[Album Details].[Artist ID]
INNER JOIN [STUDENTS\sd0815].[Artist Details] ON [STUDENTS\sd0815].[Album Details].[Artist ID] =
[STUDENTS\sd0815].[Artist Details].[Artist ID ]
INNER JOIN [STUDENTS\sd0815].[Production Company] ON [STUDENTS\sd0815].[Artist Details].Label =
[STUDENTS\sd0815].[Production Company].[Company ]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```

SELECT DISTINCT [STUDENTS\sd0815].Song.[Song Name], [STUDENTS\sd0815].Acoustics.[Song Duration in ms], [STUDENTS\sd0815].Album.[Album Name],
[STUDENTS\sd0815].Artist Details.[Artist Name], [STUDENTS\sd0815].Production Company.[Company]
FROM [STUDENTS\sd0815].Song
INNER JOIN [STUDENTS\sd0815].Acoustics ON [STUDENTS\sd0815].Song.[Song ID] = [STUDENTS\sd0815].Acoustics.[Song ID]
INNER JOIN [STUDENTS\sd0815].Album Details ON [STUDENTS\sd0815].Song.[Artist ID] = [STUDENTS\sd0815].Album Details.[Artist ID]
INNER JOIN [STUDENTS\sd0815].Artist Details ON [STUDENTS\sd0815].Album Details.[Artist ID] = [STUDENTS\sd0815].Artist Details.[Artist ID]
INNER JOIN [STUDENTS\sd0815].Production Company ON [STUDENTS\sd0815].Artist Details.[Label] = [STUDENTS\sd0815].Production Company.[Company]

```

The Results pane shows the following data:

	Song Name	Song Duration in ms	Album	Artist Name	Company
1	Baby One More Time	211160	Baby one more time	Britney Spears	Jive
2	Attention	211475	VoiceNotes	Charlie Puth	Atlantic
3	Centuries	228360	American Psycho	Fall Out Boy	Island
4	Chantaje	195613	El Dorado	Shakira	Ace
5	Cold	234308	Red Pill Blues	Maroon 5	Interscope
6	Cool for the Summer	214740	Confident	Demi Lovato	Island
7	Counting Stars	257840	Native	One Republic	Interscope
8	Despacito - Remix	228827	Vida	Luis Fonsi	DefJam
9	Despacito (Featuring Daddy Yankee)	228200	Vida	Luis Fonsi	DefJam
10	E.T. -feat. Kanye West	229573	Teenage Dream	Katy Perry	Capitol
11	Get Lucky	369627	Random Access Memories	Daft Punk	Columbia

The status bar at the bottom indicates: Query executed successfully. bcs5420.cob.internal.unt.e... STUDENTS\sd0815 (55) sd0815 00:00:00 39 rows

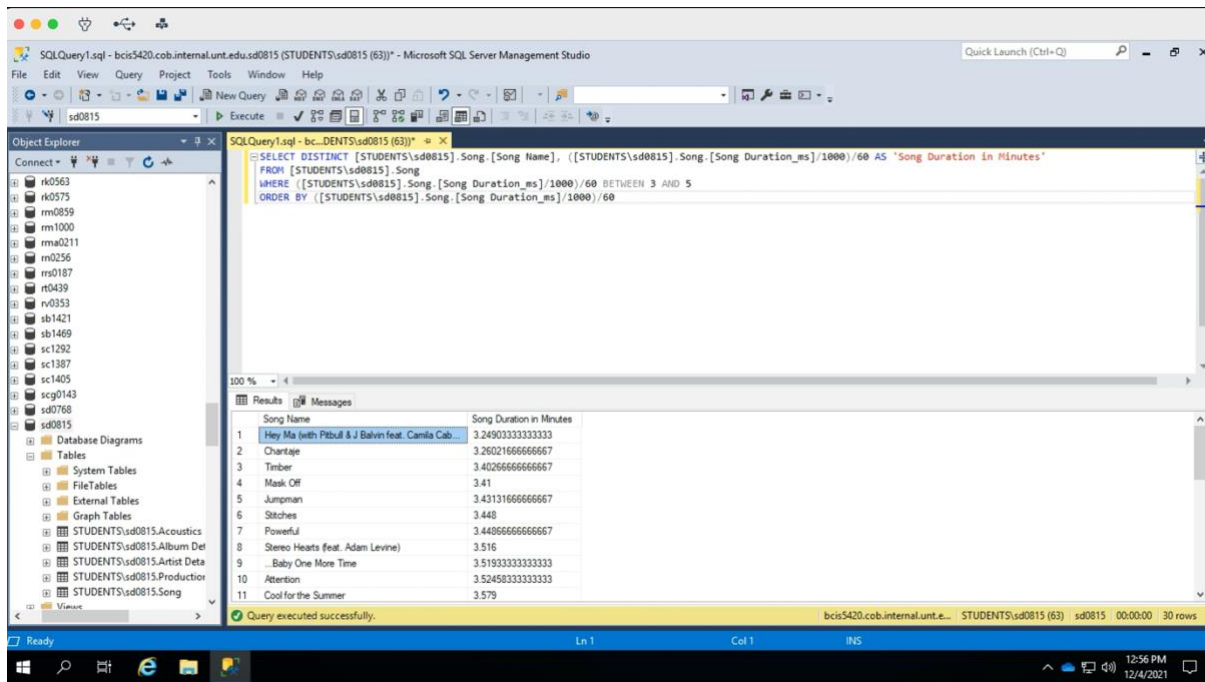
In this query, we first use select distinct to select the distinct values of the column names specified in the question. We first join the table Song with the table Acoustics using the Column Song ID with Inner join. We then joined the Song table with the Album details table with Artist ID using Inner join. Again, we joined Artist details with Album details using Artist ID with Inner Join. Finally, we joined Artist Details with Production Company with Inner Join using Company.

17. List out the number of songs with song duration between 3 mins and 5 mins and order in Ascending order. – If someone wants songs based on Song duration.

```

SELECT DISTINCT [STUDENTS\sd0815].Song.[Song Name], ([STUDENTS\sd0815].Song.[Song
Duration_ms]/1000)/60 AS 'Song Duration in Minutes'
FROM [STUDENTS\sd0815].Song
WHERE ([STUDENTS\sd0815].Song.[Song Duration_ms]/1000)/60 BETWEEN 3 AND 5
ORDER BY ([STUDENTS\sd0815].Song.[Song Duration_ms]/1000)/60

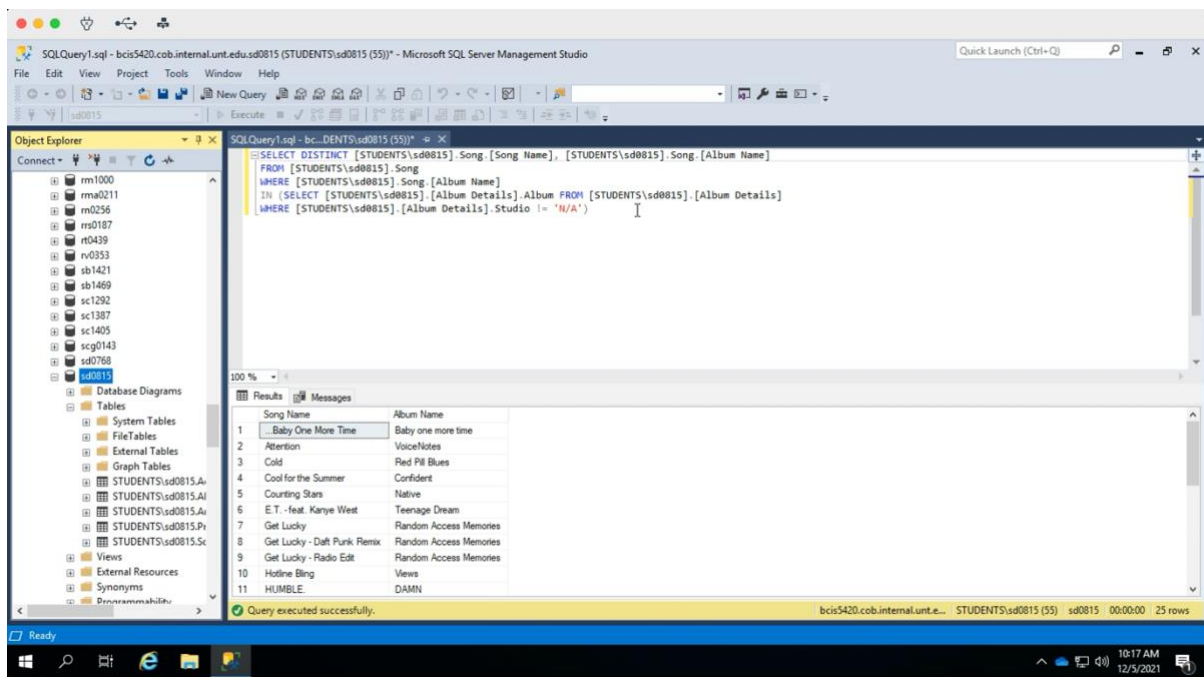
```



In this query we select the columns song name and then convert microseconds to minutes using the formula  $([STUDENTS\sd0815].Song.[Song\ Duration\_ms]/1000)/60$  and renaming it as 'Song Duration in Minutes' and selecting songs which lie between 3 mins and 5 mins and then use order by to order them in ascending order.

18. Find the list of songs which have been recorded in a studio.- If someone wants song details based on Album details.

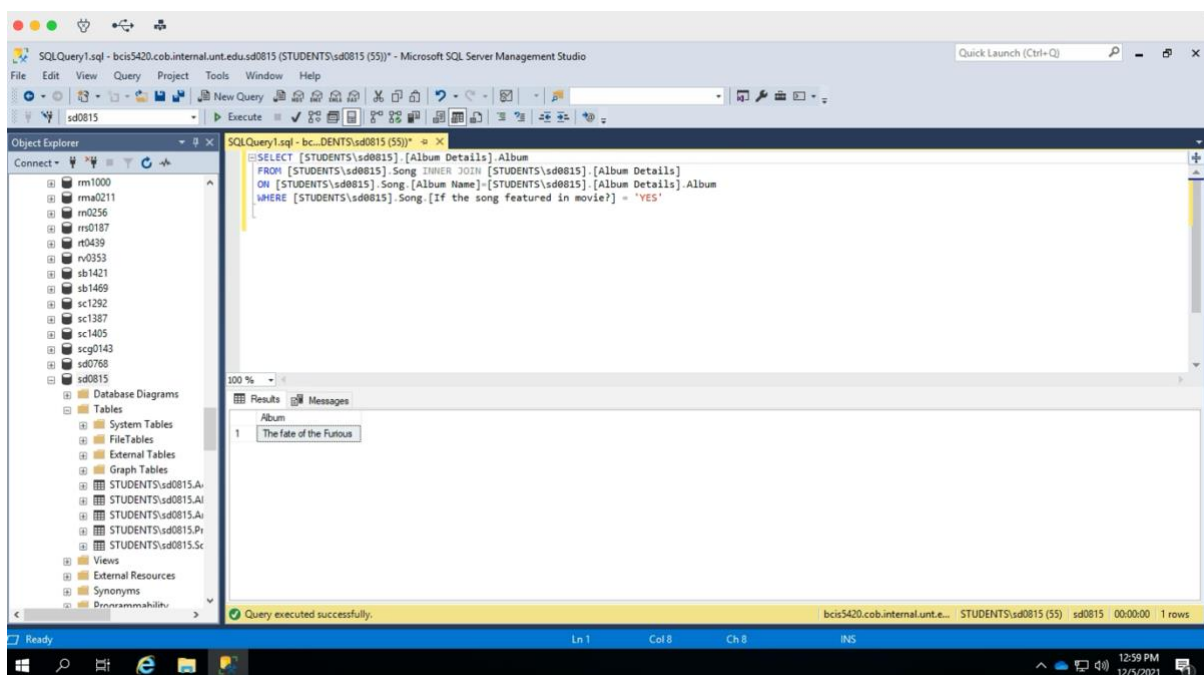
```
SELECT DISTINCT [STUDENTS\sd0815].Song.[Song Name], [STUDENTS\sd0815].Song.[Album Name]
FROM [STUDENTS\sd0815].Song
WHERE [STUDENTS\sd0815].Song.[Album Name]
IN (SELECT [STUDENTS\sd0815].[Album Details].Album FROM [STUDENTS\sd0815].[Album Details]
WHERE [STUDENTS\sd0815].[Album Details].Studio != 'N/A')
```



In this Query we used the distinct operators to avoid repetitive selection of songs. We select the columns song name and album name from the table song and then use a where condition and select the column album name and use the sub-query and select the album from the album details table and specifying the condition that studio!= 'N/A'.

19. List the name of the albums in which songs have been featured in a movie - If some wants song list based on if the song has been featured in a movie.

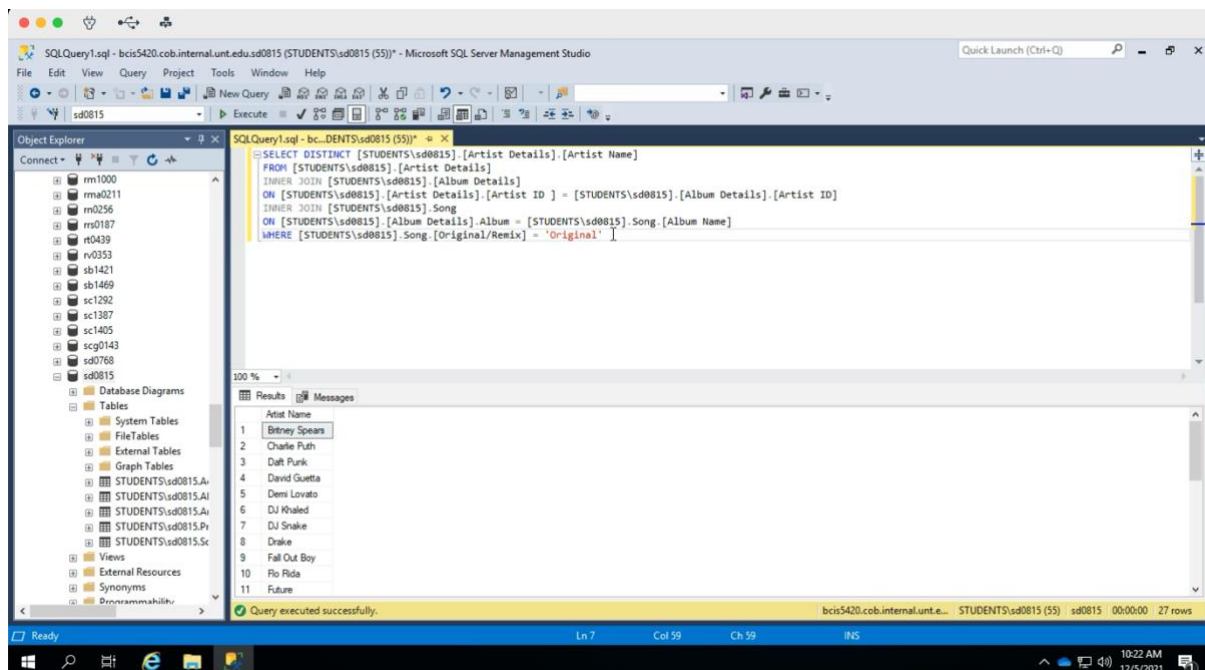
```
SELECT [STUDENTS\sd0815].[Album Details].[Album]
FROM [STUDENTS\sd0815].Song INNER JOIN [STUDENTS\sd0815].[Album Details]
ON [STUDENTS\sd0815].Song.[Album Name]= [STUDENTS\sd0815].[Album Details].[Album]
WHERE [STUDENTS\sd0815].Song.[If the song featured in movie?] = 'YES'
```



In this Query we select the column Album from the table Album details and inner join the table song with album details table using the common column album name and use the condition if the song featured in movie?= yes.

20. Find Artists who have songs that are original.- If someone wants songs which are Original and not remixes.

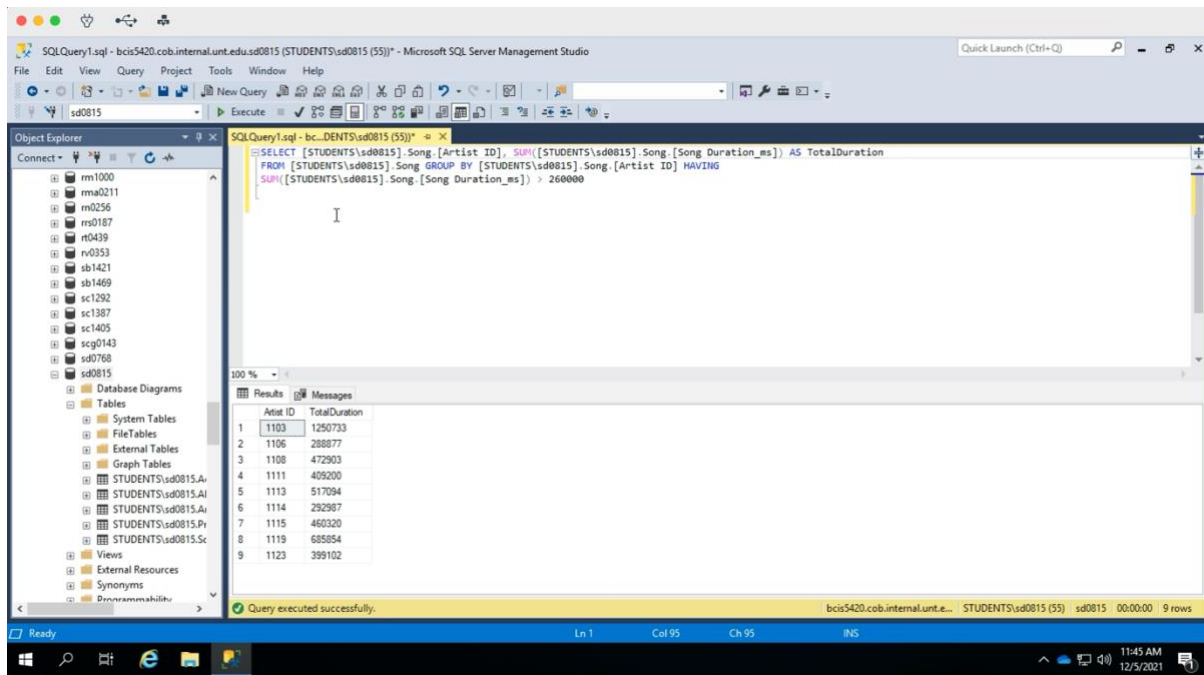
```
SELECT DISTINCT [STUDENTS\sd0815].[Artist Details].[Artist Name]
FROM [STUDENTS\sd0815].[Artist Details]
INNER JOIN [STUDENTS\sd0815].[Album Details]
ON [STUDENTS\sd0815].[Artist Details].[Artist ID ] = [STUDENTS\sd0815].[Album Details].[Artist ID]
INNER JOIN [STUDENTS\sd0815].Song
ON [STUDENTS\sd0815].[Album Details].Album = [STUDENTS\sd0815].Song.[Album Name]
WHERE [STUDENTS\sd0815].Song.[Original/Remix] = 'Original'
```



In this Query we first select the column names artist name from the table artist details and inner join album details table using the common column artist id and again join the table song using the common column album name and use the where condition to find songs that are original.

21. List out the Artist Name and Song duration, where the time duration of all songs combined is more than 260000 ms. – If someone wants songs based on total song duration.

```
SELECT [STUDENTS\sd0815].Song.[Artist ID], SUM([STUDENTS\sd0815].Song.[Song Duration_ms]) AS
TotalDuration
FROM [STUDENTS\sd0815].Song GROUP BY [STUDENTS\sd0815].Song.[Artist ID] HAVING
SUM([STUDENTS\sd0815].Song.[Song Duration_ms]) > 260000
```



In this Query, we select the columns Artist ID and song duration in milliseconds and name it as Total Duration from the table Song and use the group by condition to group the rows with the same Artist ID together. We then use having to specify the condition for the group by clause, which is that total duration is greater than 260000.

### **Conclusion:**

In this project, we have learnt how to extract data from large datasets, which would be useful for gathering multiple rows of information within seconds.

### **References:**

1. <https://www.kaggle.com/aeryan/spotify-music-analysis/notebook>
2. <https://www.google.co>
3. <https://en.wikipedia.org/>