***Analytical SQL Case Study Project***

*10 Queries are used in this case*

*Datasets Used:*

* *Songs Dataset*
* *Events Dataset*

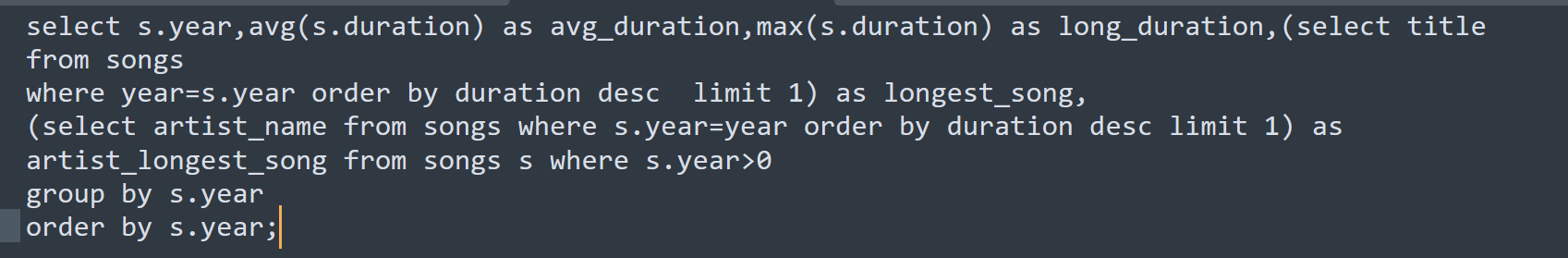
*Notes:*

* *This project is based on SQL queries and database**management, focusing on retrieving, organizing, and analyzing data from a songs dataset.*

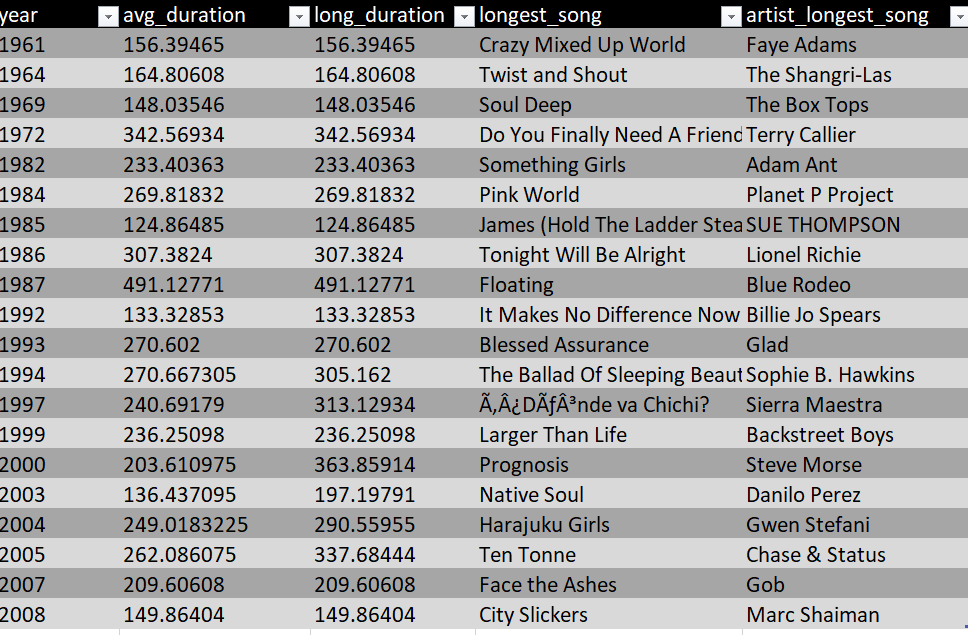
*The queries demonstrate the use of:*

* *Filtering data using WHERE conditions*
* *Grouping and aggregation with GROUP BY, SUM(), AVG(), COUNT()*
* *Conditional logic using CASE*
* *Self-joins and subqueries to handle complex relationships*
* *Sorting and limiting results for better presentation*

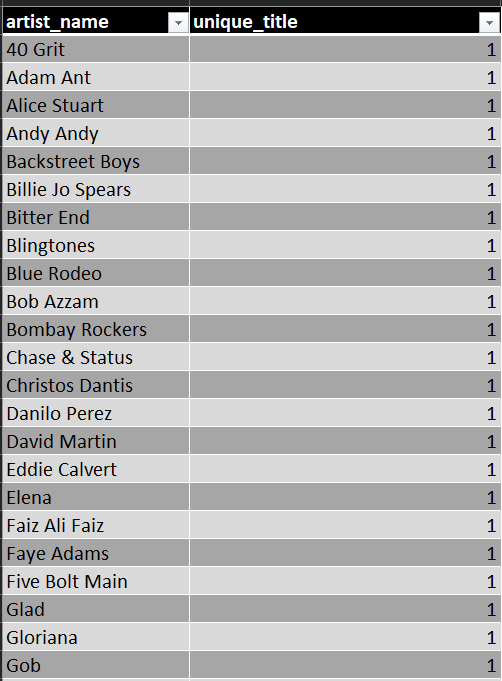
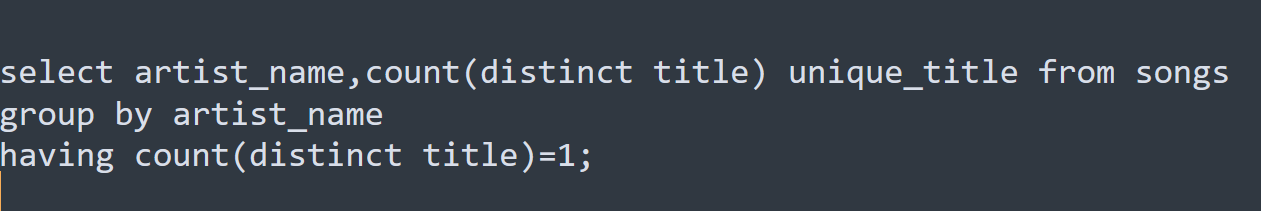
**Query 1:**

For each year, find the average duration of songs and also identify the longest song (with its artist) released in that year

* This query analyzes songs on a yearly basis. It calculates the average duration of songs, the longest duration, and also identifies the longest song title with its artist for each year. By using aggregate functions (AVG, MAX) along with correlated subqueries, we not only summarize the data but also link it with actual song details. This gives a clear picture of music trends across different years.

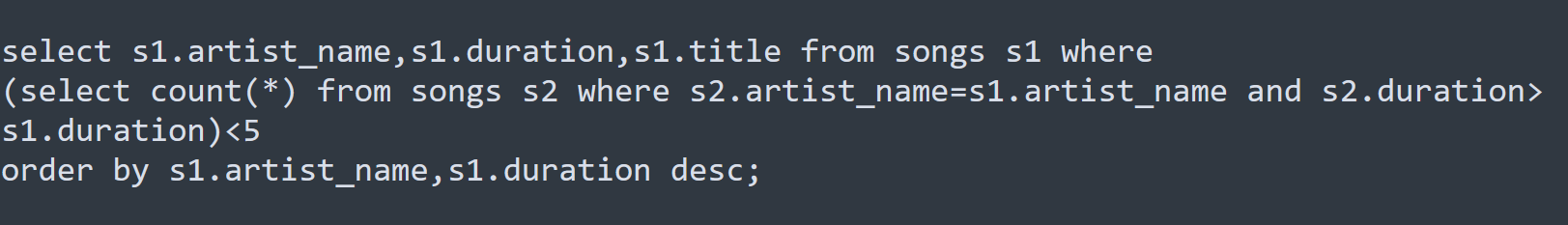
**

***QUERY 2:***

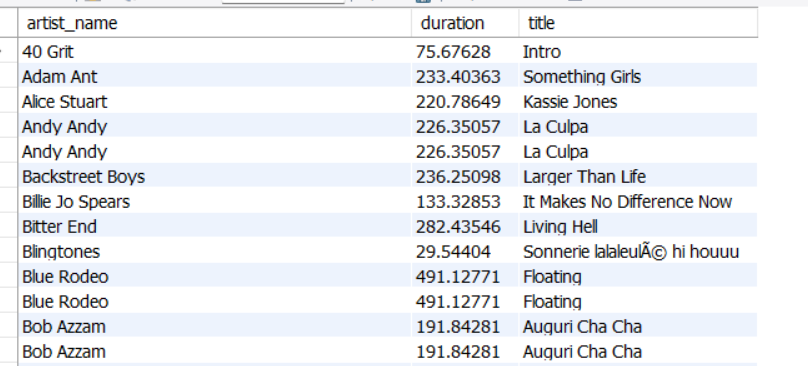
*****Find artists who have only 1 unique song in the dataset.*

* This query finds artists who have released only one unique song in the dataset.

***QUERY 3***

*******Rank the top 5 longest songs for each artist using a subquery.*

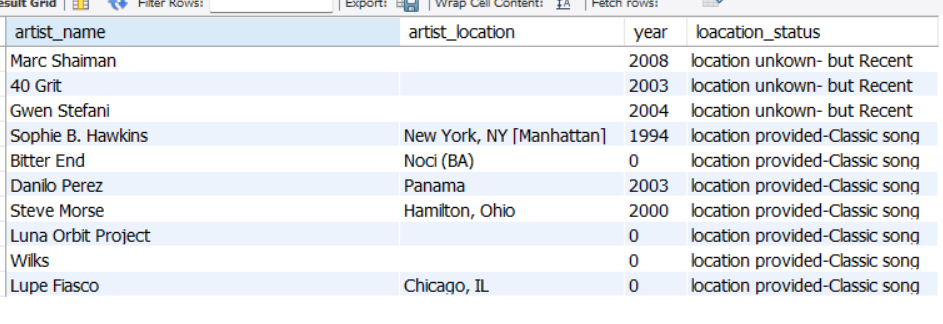
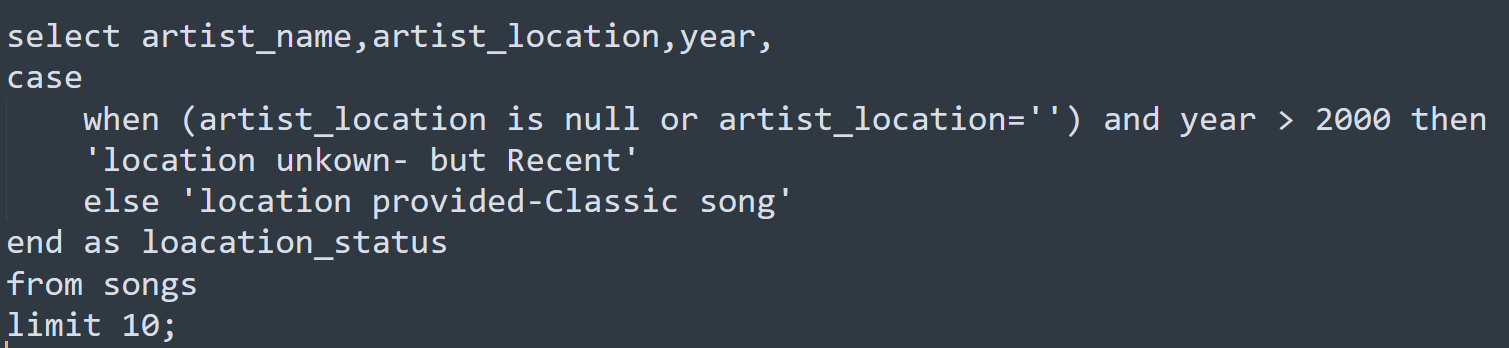
* This query returns the top 5 longest songs (by duration) foreach artist, ordered by artist and duration.



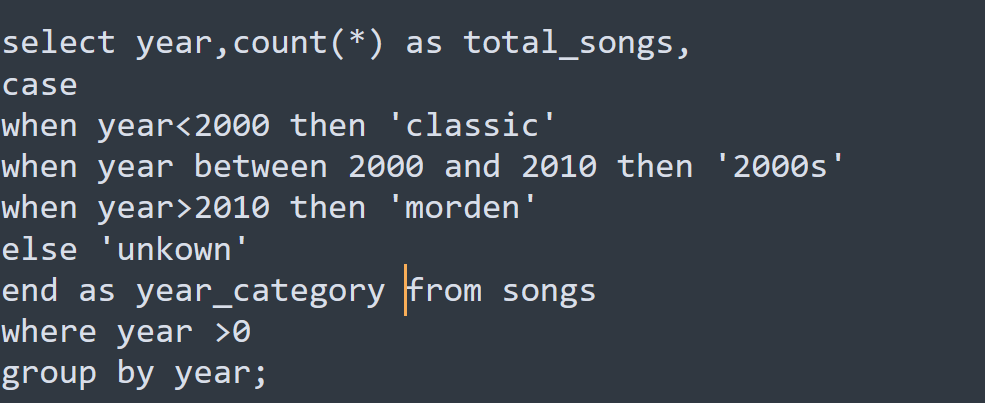
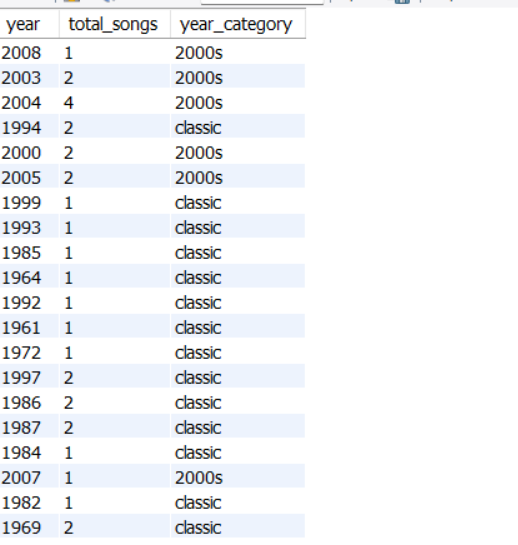
***QUERY 4:***

*Classify 10 each song into categories based on whether the artist's location is provided or missing. Additionally,*

*if the location is missing and the song was released after the year 2000, label it as "Location Unknown but Recent". Display the title,artist name, year, and the artist-location.*

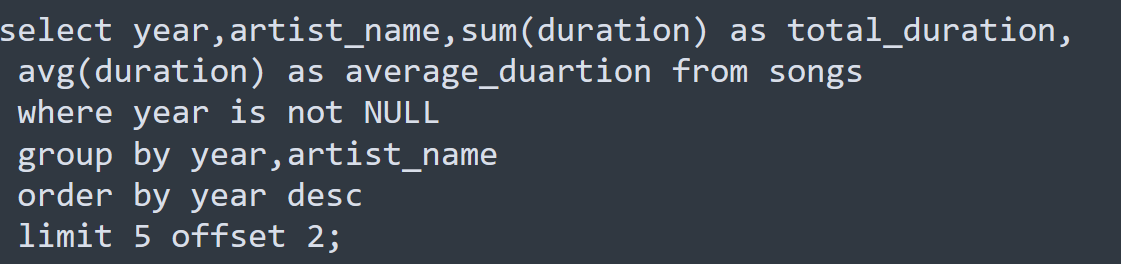
**

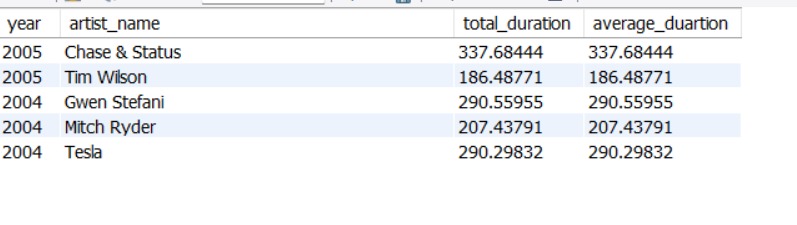
***QUERY 5:***

Write a query to analyze the songs based on their release year.The query should return the year, the total number of songs released in that year,and also classify each year into categories.If the year is before 2000,it should be labeled as Classic. If the year falls between 2000 and 2010, it should be categorized as 2000s. If the year is after 2010, it should be labeled as Modern. For any other cases, it should be shown as Unknown. this will help in understanding the trend of songs across different time periods.

* The query shows how many songs were released per year and classifies each year into Classic, 2000s, or Modern.

***QUERY 6:***

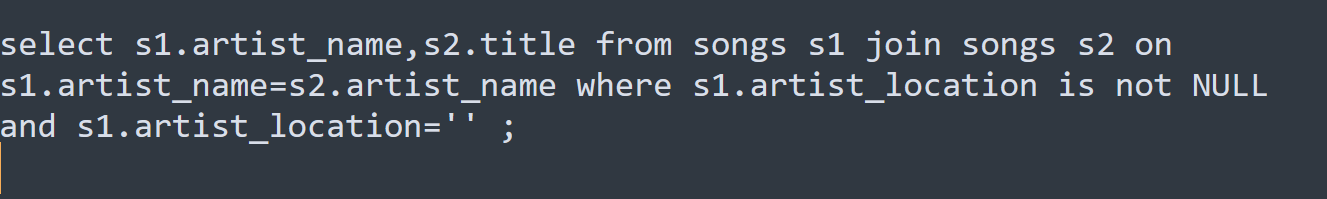
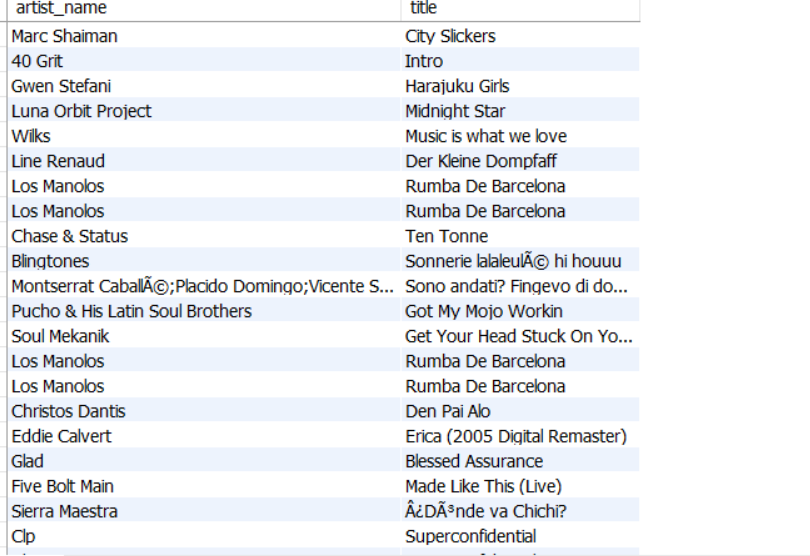
Write a query to calculate the total and average song duration for each artist in every release year.Display the 5 results sorted by release year in descending order, while skipping the first two records.

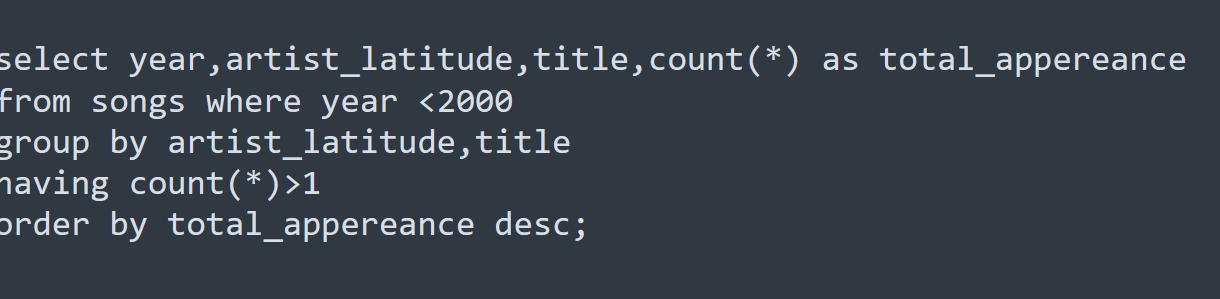


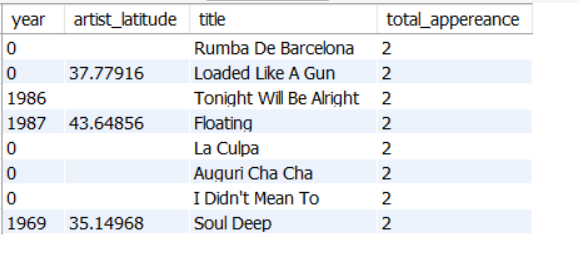
* The query shows each artist’s total and average songduration per year, sorted by most recent years, and dis

plays **5** rows after skipping the first 2.

***QUERY 7:***

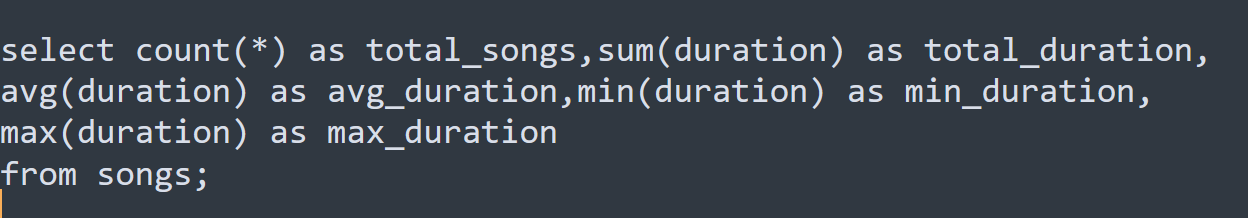
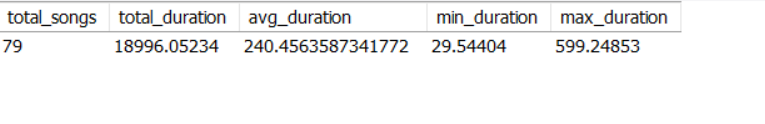
Write a SQL query to display the names of artists along with the titles of their songs, but only for those artists whose names are not missing or empty. Use a self join on the songs table to achieve this.***QUERY 8:***

Write a SQL query to find the songs released before the year 2000 that appear more than once for the same artist\_latitude. Display the release year, artist latitude, song title, and total number of appearances, and sort the results by total appearances in descending order.

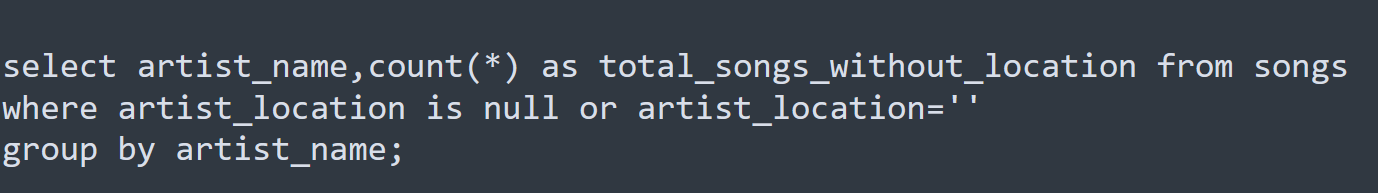


* This query finds duplicate songs (before 2000**)** for each artist’s latitude, showing year, latitude, title, and how many times they appear, sorted by most frequent first.

***QUERY 9:***

Find the total number of songs, their total duration, the average duration, the shortest song duration, and the longest song duration from the dataset.

***QUERY 10:***

Which artists have missing location information, and how many songs do they have in the dataset

