**Music Store Data Analysis**

**ABSTRACT:**

The music retail sector is leveraging data analysis to gain valuable insights for informed decision-making and operational optimization, as music stores now serve as hubs for data generated from various aspects of their operations.

The Music Store Data Analysis project explores a music store's database to uncover patterns, trends, and insights. It includes tables like 'employee,' 'customer,' 'invoice,' 'track,' 'playlist,' 'artist,' 'album,''media\_type,' and 'genre.' These tables contain information about employees, customers, sales transactions, music tracks, playlists, artists, albums, and genres.

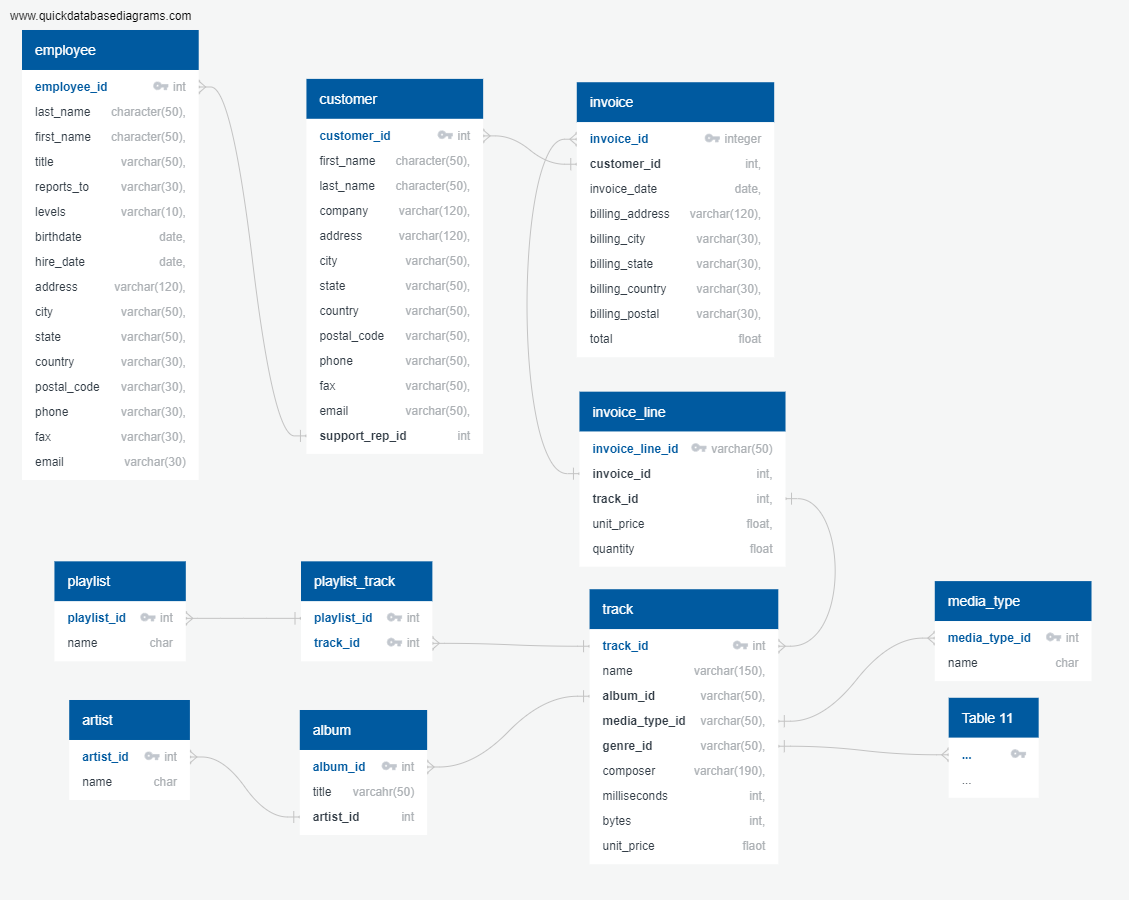
**AIM:**

This project aims to provide data-driven insights to a music store, enabling strategic decisions and enhancing customer satisfaction and profitability. The study identifies senior employees, top invoice-paying countries, top music genres, and top spending customers by analyzing job titles, customer preferences, and purchase data, while also revealing popular music genres in each country.

**Concept Used:**

SQL Database Management:The provided schema will be utilized for data extraction, transformation, and analysis using SQL queries.

**Database Schema:**

****

**Table Details:**

* Employee Table
* Customer Table
* Invoice Table
* Invoice\_line Table
* Track Table
* Album Table
* Artist Table
* Playlist Table
* Playlist\_track Table
* Media\_type Table
* Genre Table

[Use the link to access table details.](https://drive.google.com/drive/folders/1OsHhLx8RW8xHzxVFG20IWwORRmCQPBDH?usp=sharing)

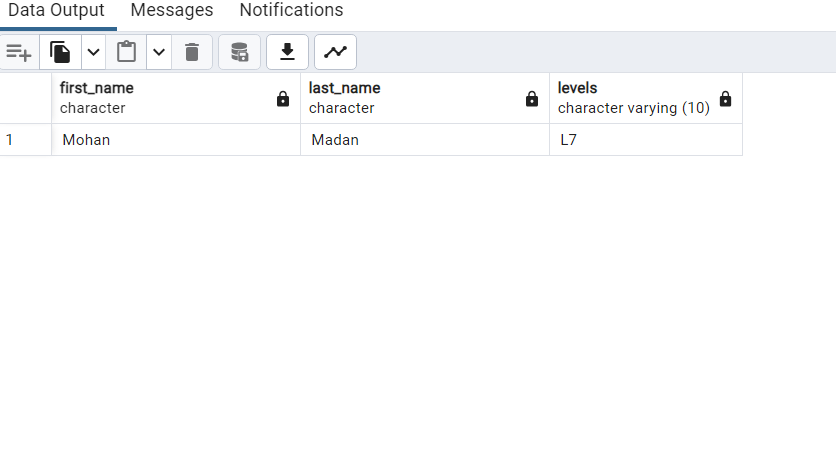
**Data Insights**

1. **Who is the senior most employee based on job title?**

**select** first\_name, last\_name, levels from employee

**Order by** levels desc

**limit** 1



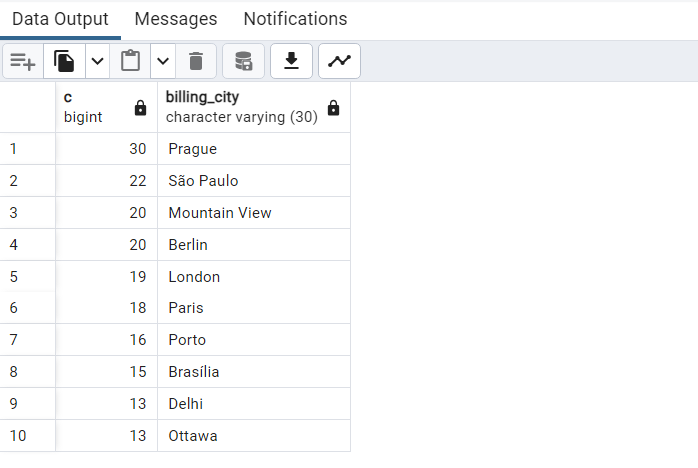
1. **Which countries have the most Invoices?**

**select** count(\*) as c, billing\_city from invoice

**Group by** billing\_city

**order by** c desc

**limit** 10

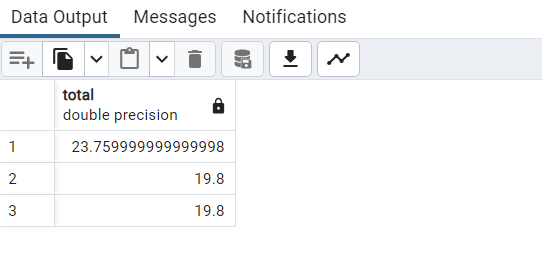


1. **What are the top 3 values of the total invoice?**

**select** total from invoice

**order by** total desc

**limit** 3



1. **Which city has the best customers? We would like to throw a promotional Music Festival in the city where we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals.**

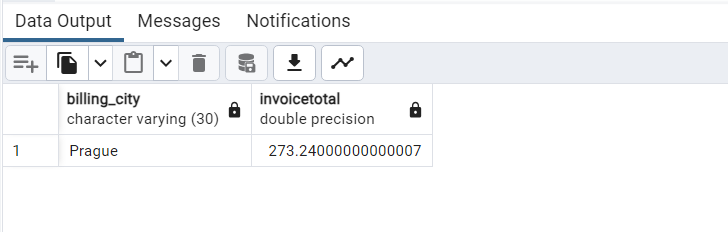
**select** billing\_city, sum(total) as InvoiceTotal

**from** invoice

**Group by** billing\_city

**order by** InvoiceTotal Desc

**Limit** 1



1. **Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money.**

**select** c.customer\_id,c.first\_name,c.last\_name, sum(total) as Total

**from** customer as c

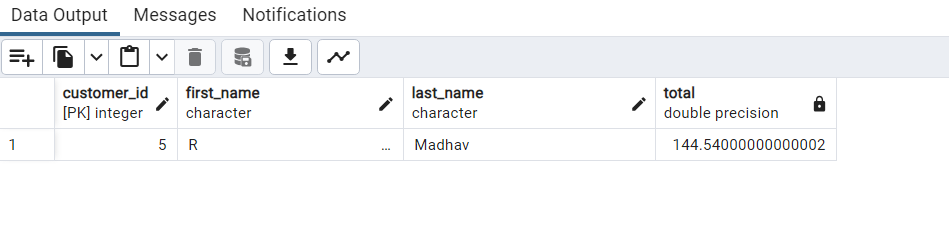
**join** invoice **as** inv

**on** c.customer\_id=inv.customer\_id

**group by** c.customer\_id

**order by** Total **desc**

**limit** 1



1. **Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A.**

**select** DISTINCT email, first\_name, last\_name

**from** customer

**join** invoice **on** customer.customer\_id=invoice.customer\_id

**join** invoice\_line **on** invoice.invoice\_id=invoice\_line.invoice\_id

**where** track\_id in(

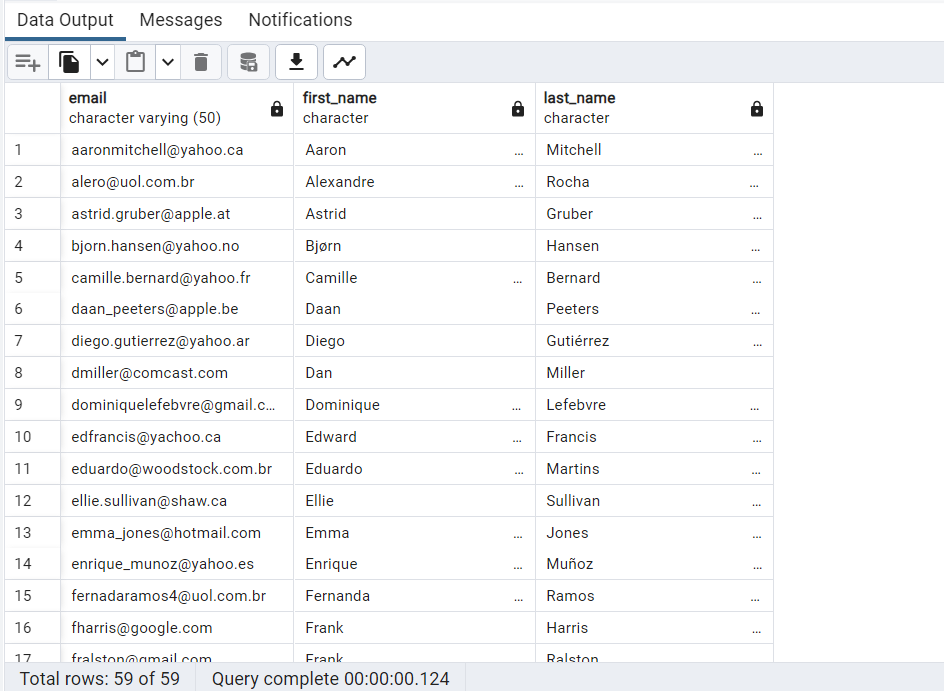
**select** track\_id from track

**join** genre **on** track.genre\_id=genre.genre\_id

**where** genre.name like 'Rock'

)

**order by** email



1. **Let us invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands.**

**select** artist.artist\_id, artist.name, count(artist.artist\_id) **as** total\_song

**from** track

**join** album **on** album.album\_id=track.album\_id

**join** artist **on** artist.artist\_id=album.artist\_id

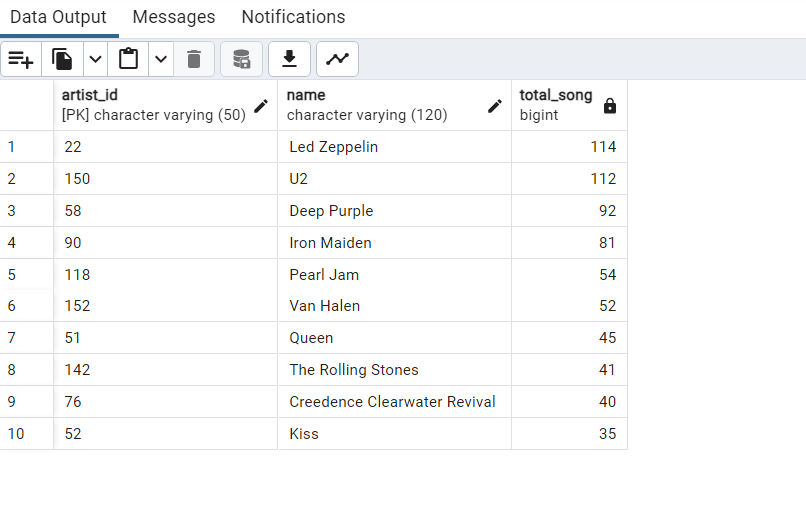
**join** genre **on** genre.genre\_id= track.genre\_id

**where** genre.name **like** 'Rock'

**group by** artist.artist\_id

**order by** total\_song **desc**

**limit** 10



1. **Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.**

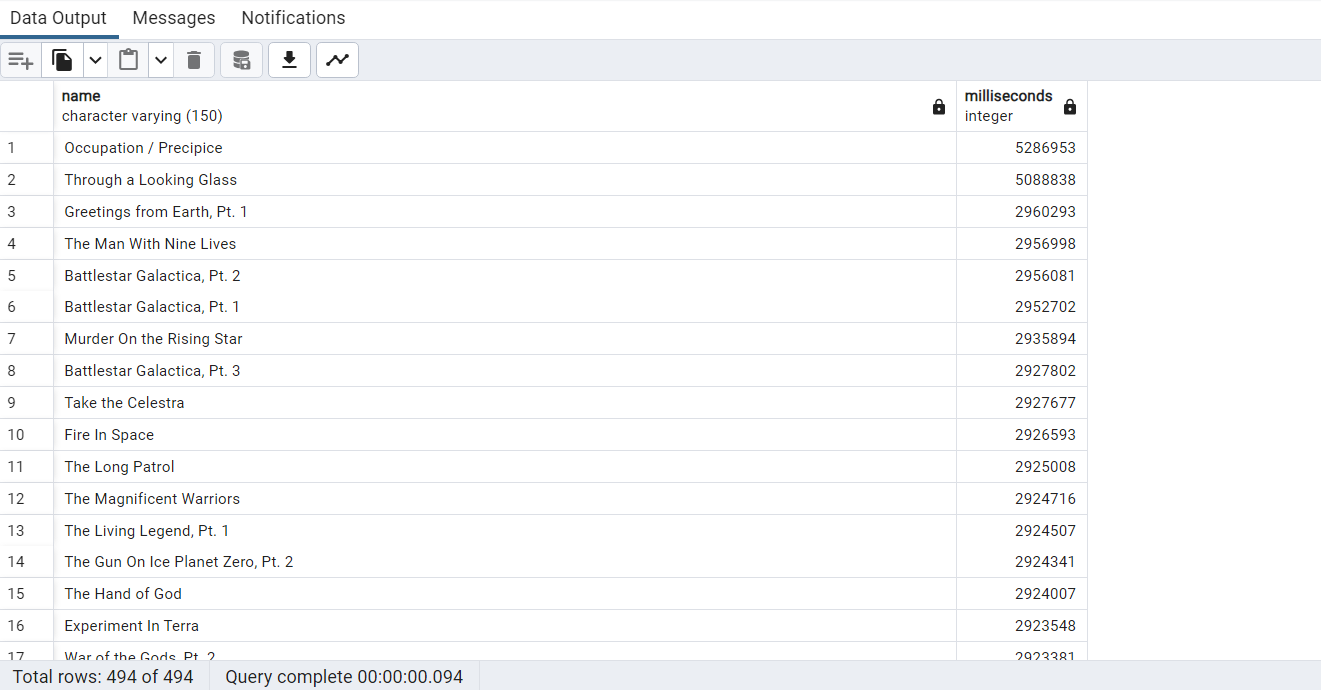
**select** name,milliseconds **from** track

**where** milliseconds > (

**select avg**(milliseconds)**from** track

)

**order by** milliseconds **desc**



1. **Find how much amount spent by each customer on artists. Write a query to return the customer’s name, artist name, and total spent**

**WITH** best\_selling\_artist **AS** (

**SELECT** artist.artist\_id, artist.name **AS** artist\_name, **SUM** (invoice\_line.unit\_price \* invoice\_line.quantity) **AS** total\_sales

**FROM** invoice\_line

**JOIN** track **ON** track.track\_id = invoice\_line.track\_id

**JOIN** album **ON** album.album\_id = track.album\_id

**JOIN** artist **ON** artist.artist\_id = album.artist\_id

**GROUP BY** 1

**ORDER BY** 3 **DESC**

**LIMIT** 1

)

**SELECT** c.customer\_id, c.first\_name, c.last\_name, bsa.artist\_name, **SUM**(il.unit\_price \* il.quantity) **AS** amount\_spent **FROM** invoice i

**JOIN** customer c **ON** c.customer\_id = i.customer\_id

**JOIN** invoice\_line il **ON** il.invoice\_id = i.invoice\_id

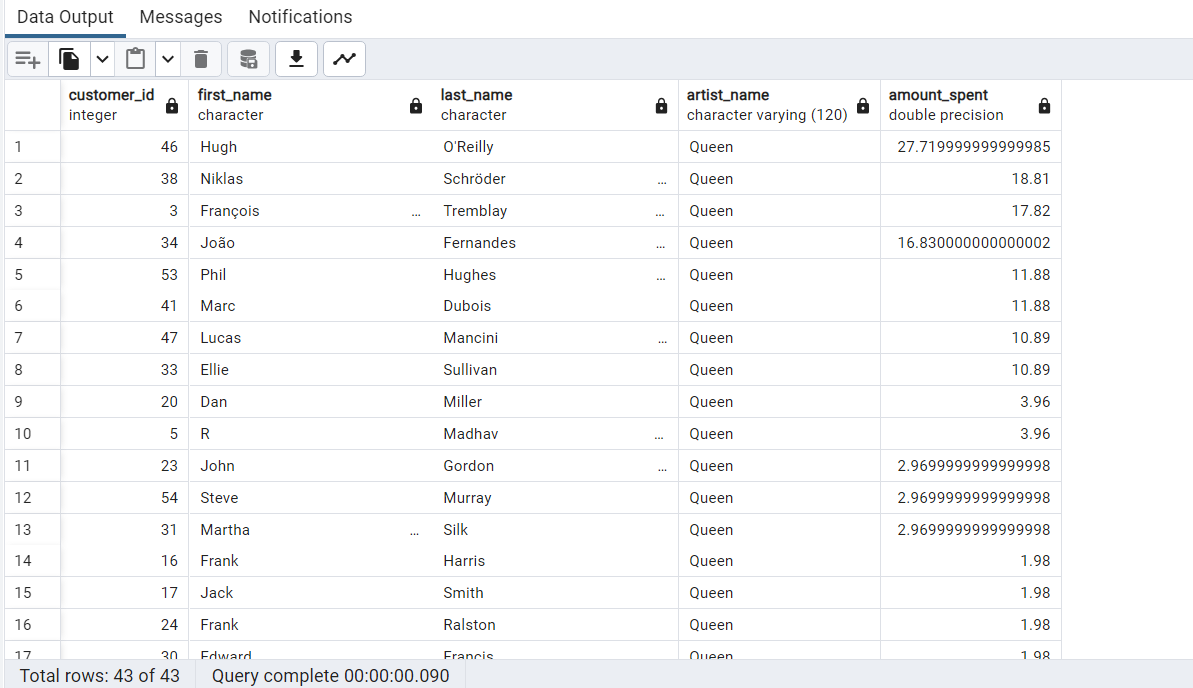
**JOIN** track t **ON** t.track\_id = il.track\_id

**JOIN** album alb **ON** alb.album\_id = t.album\_id

**JOIN** best\_selling\_artist bsa **ON** bsa.artist\_id = alb.artist\_id

**GROUP BY** 1, 2, 3, 4

**ORDER BY** 5 DESC



1. **We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest number of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres.**

**with recursive**

sales\_per\_country as (

**selec**t count(\*) as purchase\_per\_genre, customer.country, genre.name, genre.genre\_id

**from** invoice\_line

**join** invoice **on** invoice.invoice\_id = invoice\_line.invoice\_id

**join** customer **on** customer.customer\_id= invoice.customer\_id

**join** track **on** track.track\_id= invoice\_line.track\_id

**join** genre **on** genre.genre\_id= track.genre\_id

**Group by** 2,3,4

**order by** 2

),

max\_genre\_per\_country as (

**select max**(purchase\_per\_genre) as max\_genre , country

**from** sales\_per\_country

**group by** 2

**order by** 2

)

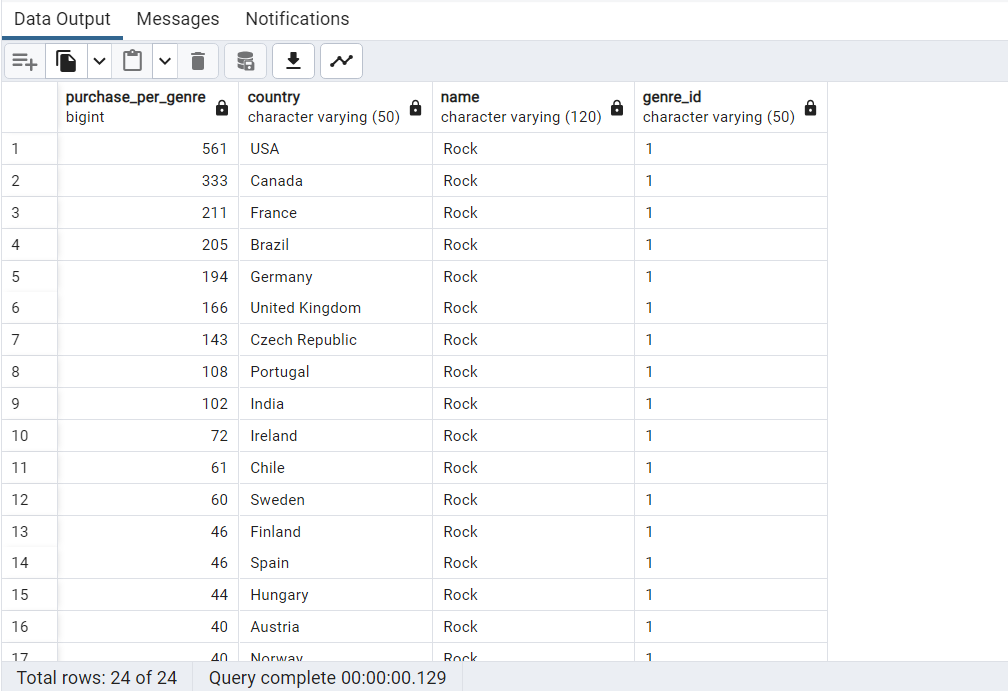
**select** sales\_per\_country .\*

**from** sales\_per\_country

**join** max\_genre\_per\_country **on** sales\_per\_country.country = max\_genre\_per\_country.country

**where** sales\_per\_country.purchase\_per\_genre = max\_genre\_per\_country.max\_genre

**order by** sales\_per\_country.purchase\_per\_genre **desc**



1. **Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.**

WITH RECURSIVE

customter\_with\_country AS (

SELECT customer.customer\_id,first\_name,last\_name,billing\_country,SUM(total) AS total\_spending

FROM invoice

JOIN customer ON customer.customer\_id = invoice.customer\_id

GROUP BY 1,2,3,4

ORDER BY 2,3 DESC),

country\_max\_spending AS(

SELECT billing\_country,MAX(total\_spending) AS max\_spending

FROM customter\_with\_country

GROUP BY billing\_country)

SELECT cc.billing\_country, cc.total\_spending, cc.first\_name, cc.last\_name, cc.customer\_id

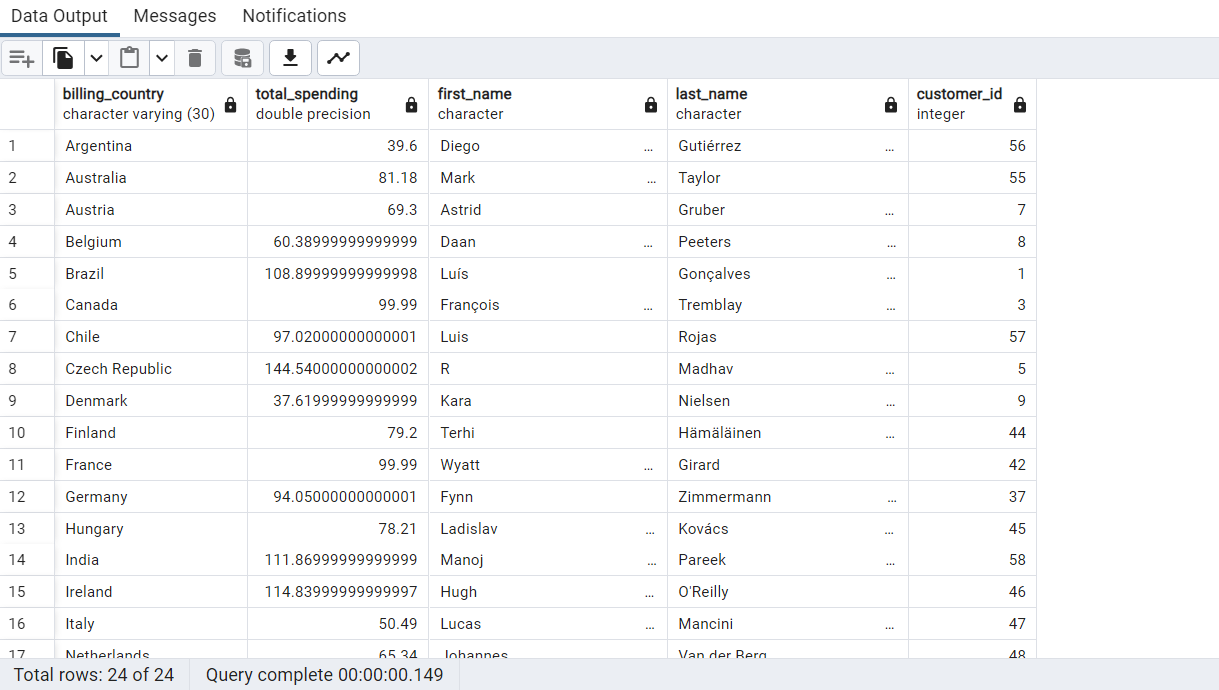
FROM customter\_with\_country cc

JOIN country\_max\_spending ms

ON cc.billing\_country = ms.billing\_country

WHERE cc.total\_spending = ms.max\_spending

ORDER BY 1;



**Conclusion**

The Music Store Data Analysis project demonstrates the importance of data-driven decision-making in the music retail industry. It analyzes employee hierarchy, customer behavior, music preferences, and genre popularity to guide strategic decisions. SQL queries provide actionable insights for optimizing operations, enhancing customer engagement, and maximizing revenue, ultimately leading to a more successful music store.