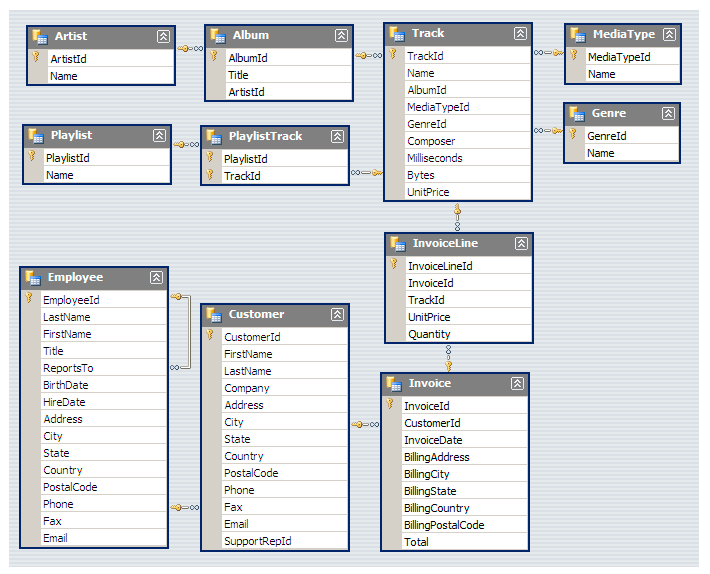
**MUSIC STORE ANALYSIS**

**OBJECTIVE**

The objective of this beginner-level SQL project for a Music Store is to design and implement a relational database and write SQL queries to explore and analyze the store's operations. Through this project, beginners will gain hands-on experience in database management, SQL query writing and data analysis with a focus on the specific requirements of a Music Store.

 **DATABASE SCHEMA**

**DATA INSIGHTS**

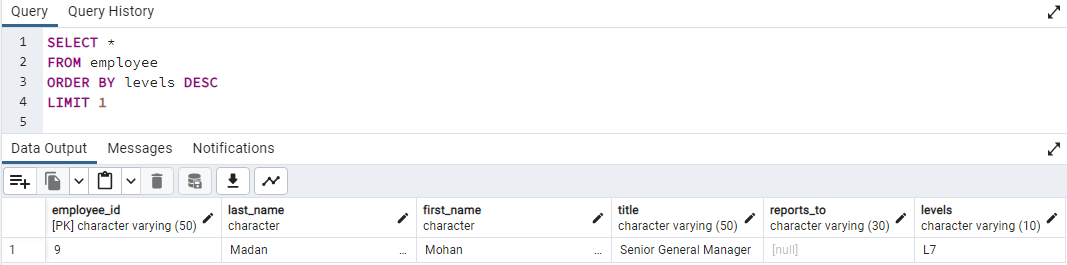
**1. Who is the senior most employee based on job title?**

SELECT \*

FROM employee

ORDER BY levels DESC

LIMIT 1



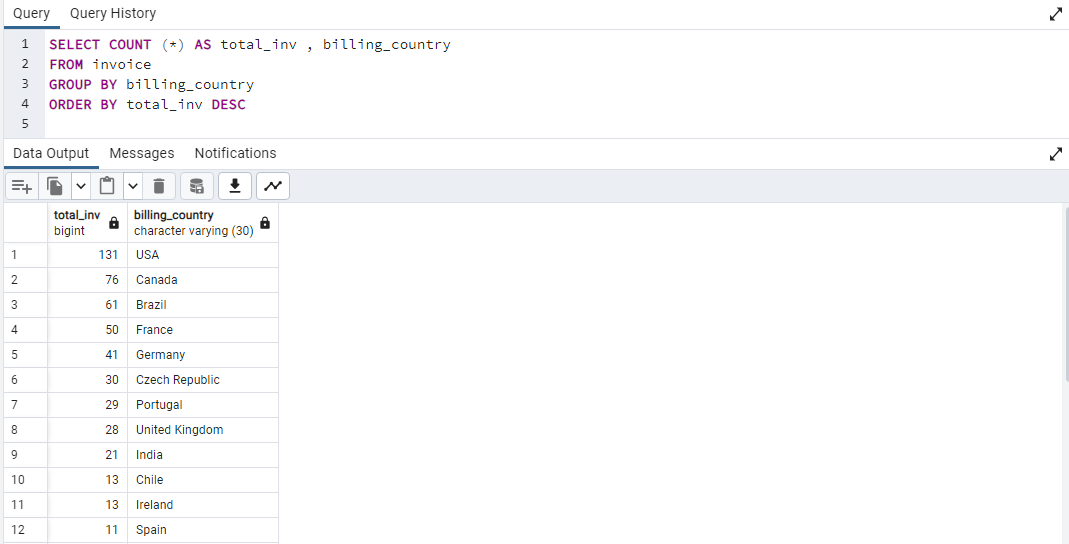
**2. Which countries have the most Invoices?**

SELECT COUNT (\*) AS total\_inv , billing\_country

FROM invoice

GROUP BY billing\_country

ORDER BY total\_inv DESC



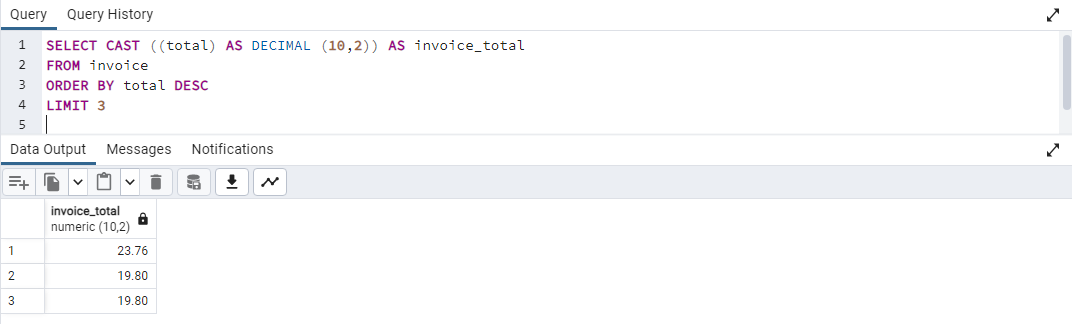
**3. What are top 3 values of total invoice?**

SELECT CAST ((total) AS DECIMAL (10,2)) AS invoice\_total

FROM invoice

ORDER BY total DESC

LIMIT 3



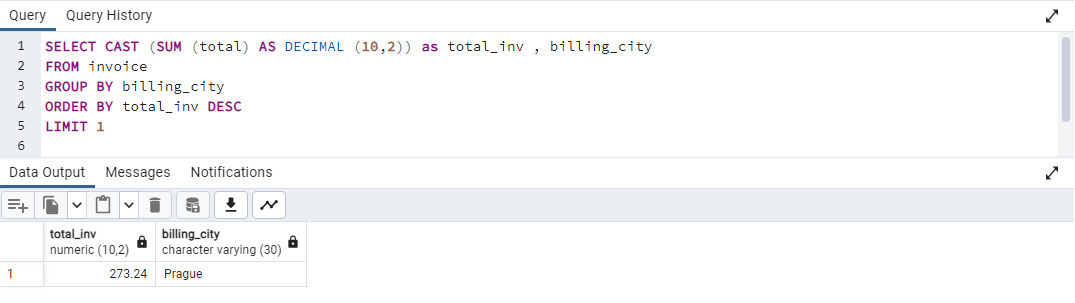
**4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money from. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals.**SELECT CAST (SUM (total) AS DECIMAL (10,2)) as total\_inv , billing\_city

FROM invoice

GROUP BY billing\_city

ORDER BY total\_inv DESC

LIMIT 1



**5. 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, CAST (SUM (i.total)AS DECIMAL (10,2)) AS total\_money\_spent

FROM invoice AS i

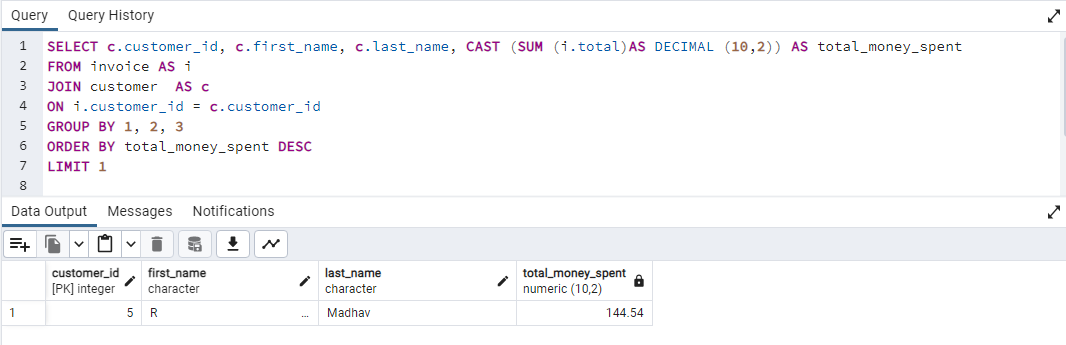
JOIN customer AS c

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

GROUP BY 1, 2, 3

ORDER BY total\_money\_spent DESC

LIMIT 1



**6. 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 c.email, c.first\_name , c.last\_name, g.name as genre\_name

FROM customer as c

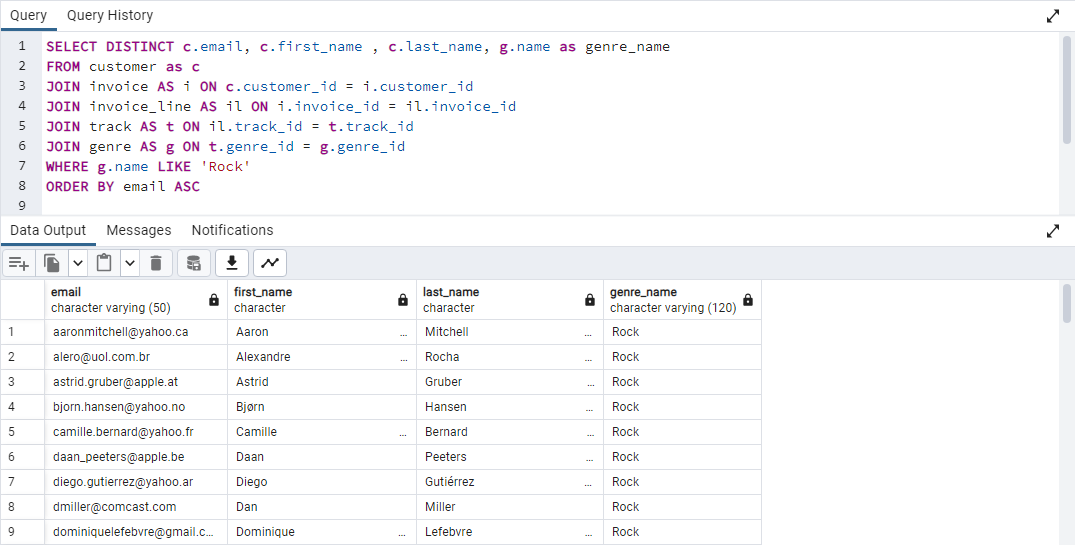
JOIN invoice AS i ON c.customer\_id = i.customer\_id

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

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

JOIN genre AS g ON t.genre\_id = g.genre\_id

WHERE g.name LIKE 'Rock'  
ORDER BY email ASC



**7. 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 a.name AS artist\_name, COUNT (t.track\_id) as track\_count

FROM artist AS a

JOIN album AS ab ON ab.artist\_id = a.artist\_id

JOIN track AS t ON t.album\_id = ab.album\_id

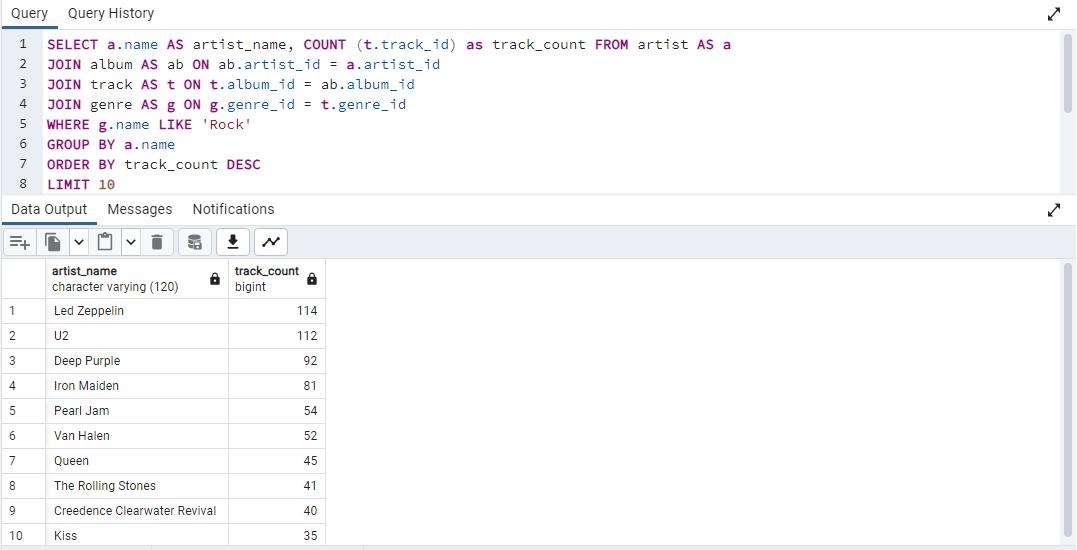
JOIN genre AS g ON g.genre\_id = t.genre\_id

WHERE g.name LIKE 'Rock'

GROUP BY a.name

ORDER BY track\_count DESC

LIMIT 10

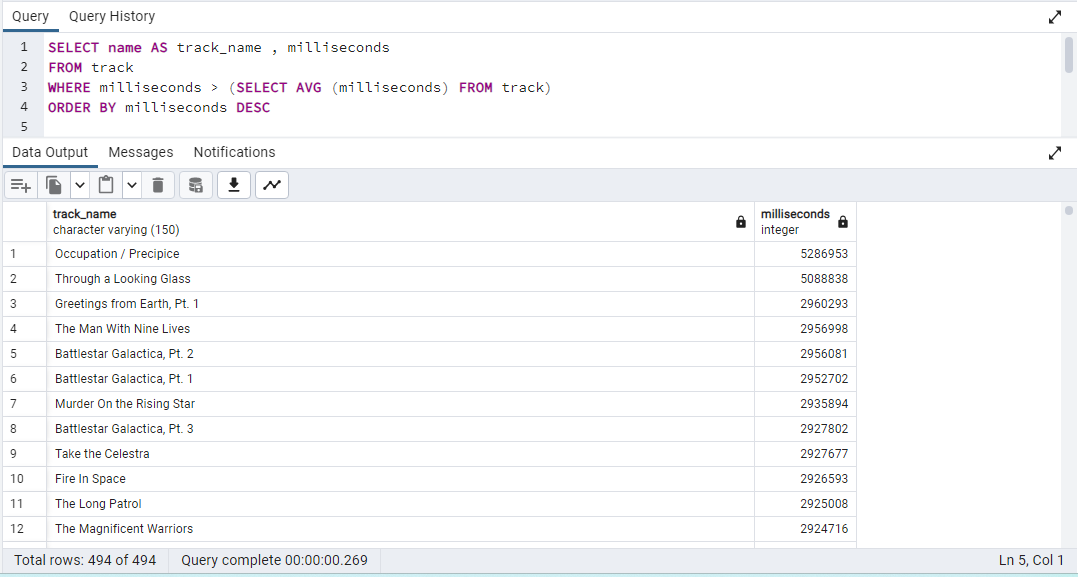


**8. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.**

SELECT name AS track\_name , milliseconds

FROM track

WHERE milliseconds > (SELECT AVG (milliseconds) FROM track)

ORDER BY milliseconds DESC  


**9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent**

SELECT CONCAT (c.first\_name ,c.last\_name) AS customer\_name,

a.name AS artist\_name ,

CAST (SUM (il.unit\_price \* il.quantity) AS DECIMAL (10,2)) AS total\_spent

FROM customer AS c

JOIN invoice AS i ON i.customer\_id = c.customer\_id

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

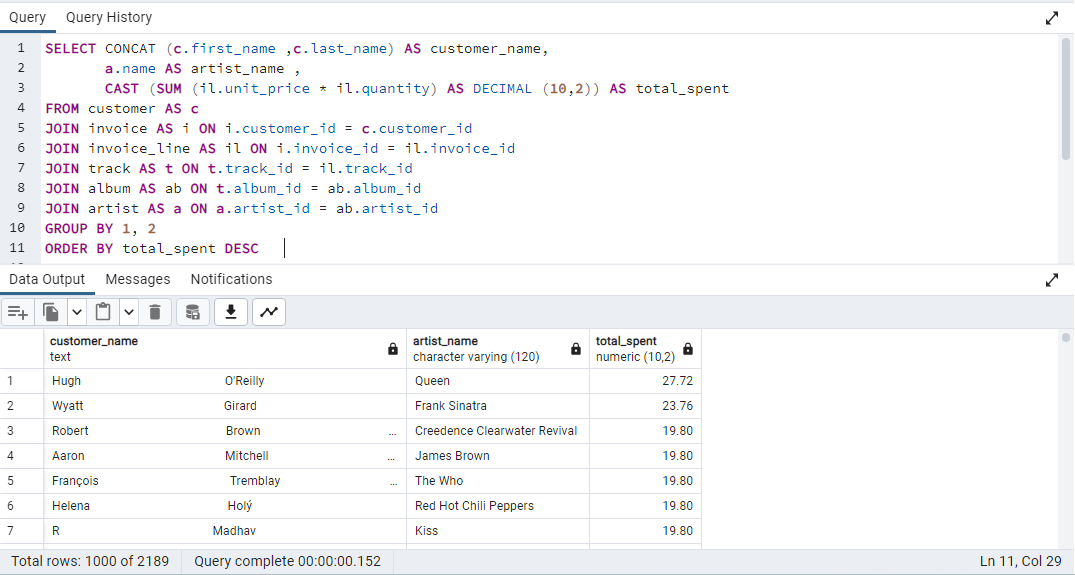
JOIN track AS t ON t.track\_id = il.track\_id

JOIN album AS ab ON t.album\_id = ab.album\_id

JOIN artist AS a ON a.artist\_id = ab.artist\_id

GROUP BY 1, 2

ORDER BY total\_spent DESC



**10. 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 popular\_genre AS

(SELECT COUNT(il.quantity) AS purchases, c.country, g.name, g.genre\_id,

ROW\_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS RowNo

FROM invoice\_line AS il

JOIN invoice AS i ON i.invoice\_id = il.invoice\_id

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

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

JOIN genre AS g ON g.genre\_id = t.genre\_id

GROUP BY 2,3,4

ORDER BY 1 DESC)

SELECT \* FROM popular\_genre WHERE RowNo <= 1



**11. 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 max\_amount\_spent AS

(SELECT c.first\_name , c.last\_name, i.billing\_country, CAST (SUM (i.total) AS DECIMAL (10,2)) AS amount\_spent,

ROW\_NUMBER () OVER (PARTITION BY i.billing\_country ORDER BY SUM (i.total) DESC) AS row\_no

FROM customer AS c

JOIN invoice AS i ON i.customer\_id = c.customer\_id

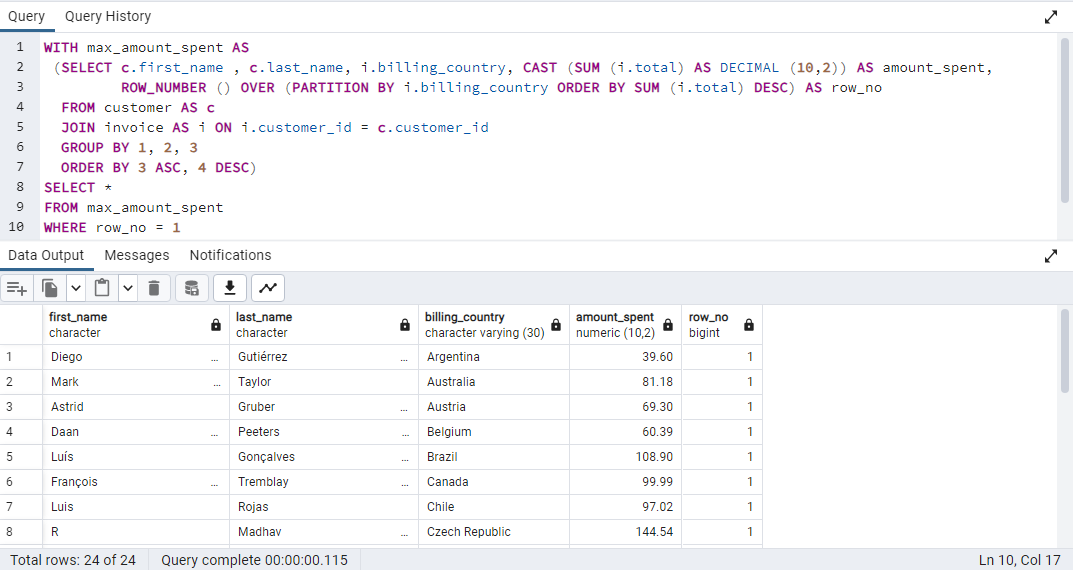
GROUP BY 1, 2, 3

ORDER BY 3 ASC, 4 DESC)

SELECT \*

FROM max\_amount\_spent

WHERE row\_no = 1



**CONCLUSION**

The Music Store Data Analysis project emphasizes the significant role of data-driven decision-making in the music retail industry. It explores employee hierarchies, customer behaviours, music preferences, and genre popularity to guide strategic decisions. With SQL queries, this project provides practical insights to optimize operations, enhance customer engagement, and maximize revenue. In essence, the project sets the stage for a more successful and efficient music store operation, showcasing the transformative power of data analysis in the industry.