**Music Store Data Analysis Project using SQL**

**Objective:** This project is for beginners and will teach you how to analyze the music playlist database. You can examine the dataset with SQL and help the store understand its business growth by answering simple question.

Application used: PostgreSQL, pg admin4

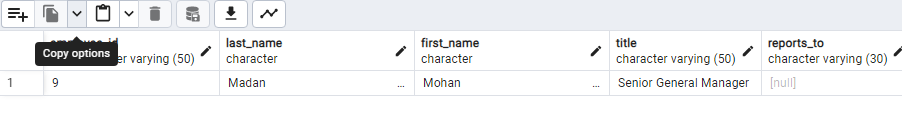
# Easy: Q 1: Who is the Senior Most Employee based on Job Title?

Ans :

Select \* from employee

ORDER BY levels desc

LIMIT 1



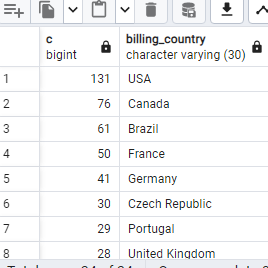
# Q 2: Which countries have the most invoices?

**Select \* from invoice;**

**Select count (\*) as c, billing\_country from invoice**

**group by billing\_country**

**order by c desc;**

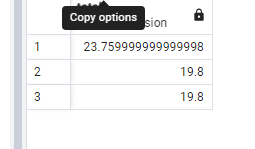


# Q3: What are top 3 Values of total invoice?

select total from invoice

order by total desc

limit 3



# Q4 : which city has the best customers ? we would like to throw a promotional Music Festival in the city we made the most money. Wrtie a query that returns one city that has the highest sum of Invoice totals. Return both the City name and sum of all invoice totals.

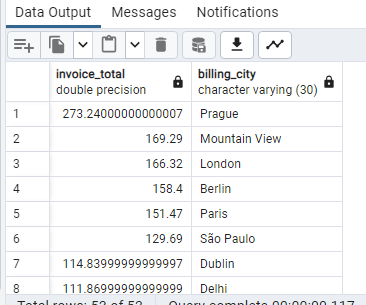
Best Customer is

select \* from invoice ;

select sum(total) as invoice\_total, billing\_city from invoice

group by billing\_city

order by invoice\_total desc



# Q5 : who is best customers ? The customer who has spent the most money will be declared the best customer. Write a query that shows the person who has spent the most money?

select customer.customer\_id, customer.first\_name, customer.last\_name, SUM(total)as total

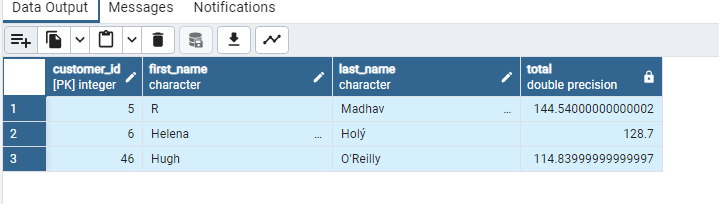
from customer

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

GROUP BY customer.customer\_id

ORDER BY total DESC

LIMIT 3



**Question Set 2 = Moderate**

**Q6. 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(**

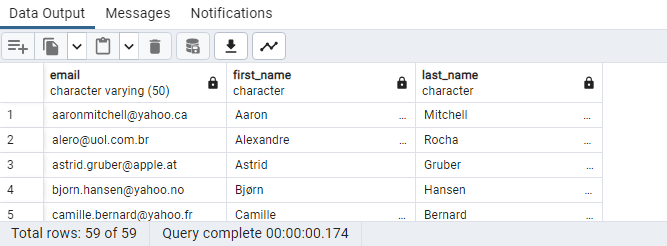
**SELECT track\_id FROM track**

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

**WHERE genre.name Like 'Rock'**

**)**

**ORDER BY email ;**



**Q7. Let’s 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 number\_of\_songs**

**FROM track**

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

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

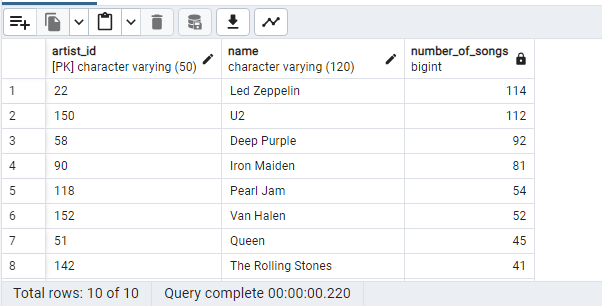
**JOIN genre ON genre.genre\_id = track.genre\_id**

**WHERE genre.name LIKE 'Rock'**

**GROUP BY artist.artist\_id**

**ORDER BY number\_of\_songs DESC**

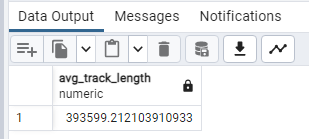
**Limit 10 ;**



**Q8. 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 AVG(milliseconds) AS avg\_track\_length

FROM track



**select \* from track**

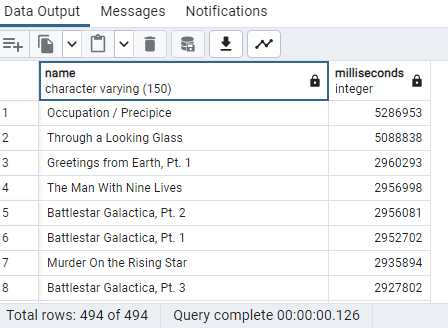
**SELECT name,milliseconds FROM track**

**WHERE milliseconds >(**

**SELECT AVG(milliseconds) AS avg\_track\_length**

**FROM track)**

**ORDER BY milliseconds DESC;**

****

**Question Set 3 : Advance**

**Q9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.**

**- customer from customer table**

**- artist from artist table**

**-Total Spent from invoice table ( unit price \* Quantity)**

**- CT (Comman Table for temp to run multiple table)**

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

**SELECT artist.artist\_id AS 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**

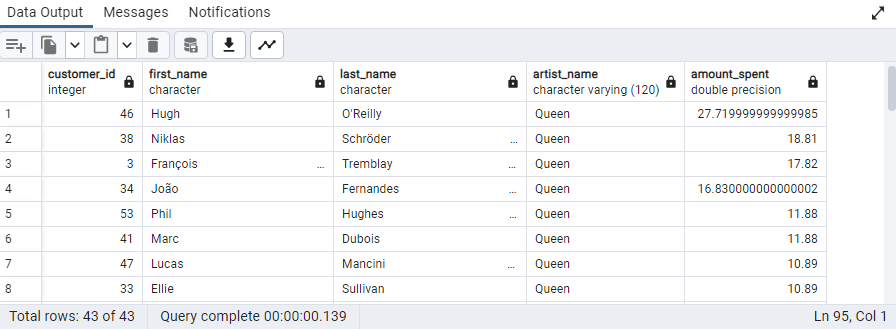
**JOIN track t ON t.track\_id = il.track\_id**

**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 ;**

****

**Q10. 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 amount 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(invoice\_line.quantity) AS purchases, customer.country,genre.name, genre.genre\_id,**

**ROW\_NUMBER() OVER (PARTITION BY customer.country ORDER BY COUNT (invoice\_line.quantity)DESC) AS RowNo**

**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**

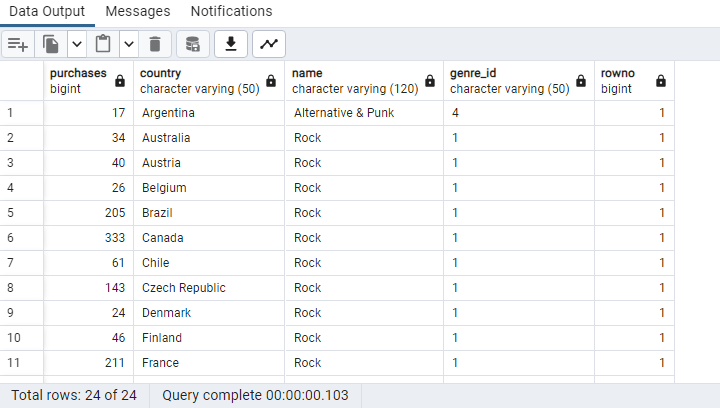
**JOIN genre ON genre.genre\_id = track.genre\_id**

**GROUP By 2,3,4**

**ORDER BY 2 ASC, 1 DESC**

**)**

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

****

**# Now same output using Recursive Mean 2nd query output is depend on 1st Query.**

**#Example 2 :**

**WITH RECURSIVE**

**Sales\_per\_country AS (**

**SELECT COUNT(\*) AS purchases\_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(purchases\_per\_genre) AS max\_genre\_number,country**

**FROM sales\_per\_country**

**GROUP BY 2**

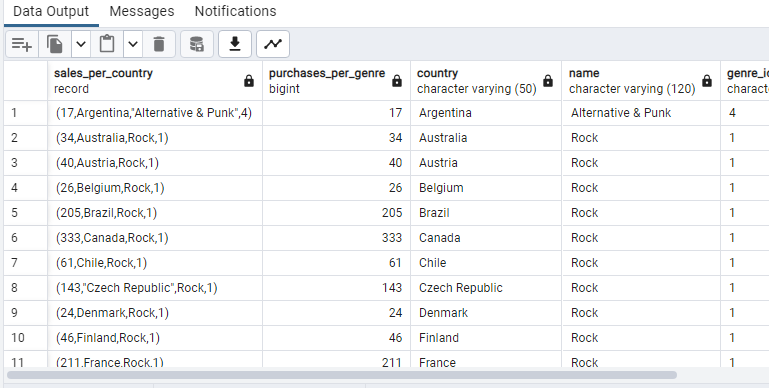
**ORDER BY 2 )**

**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.purchases\_per\_genre = max\_genre\_per\_country.max\_genre\_number**

****

**Q11.Write a query that determine the customer that has spent them ost 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**

**customer\_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 customer\_with\_country**

**GROUP BY billing\_country)**

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

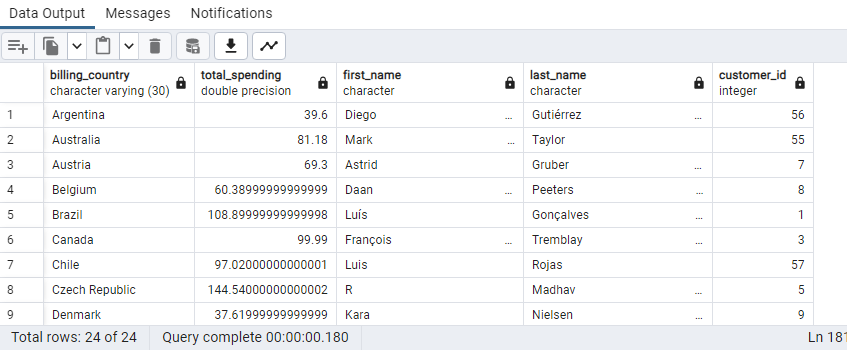
**FROM customer\_with\_country cc**

**JOIN country\_max\_spending ms**

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

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

**ORDER BY 1 ;**

****

**# Example 2 ( Using CT )**

**WITH customer\_with\_country AS (**

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

**ROW\_NUMBER() OVER(PARTITION BY billing\_country ORDER BY SUM(total) DESC) AS RowNo**

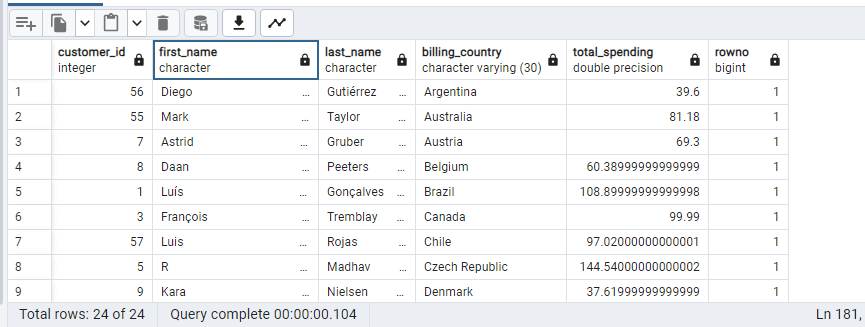
**FROM invoice**

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

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

**ORDER BY 4 ASC,5 DESC)**

**SELECT \* FROM customer\_with\_country WHERE RowNO <= 1**

****