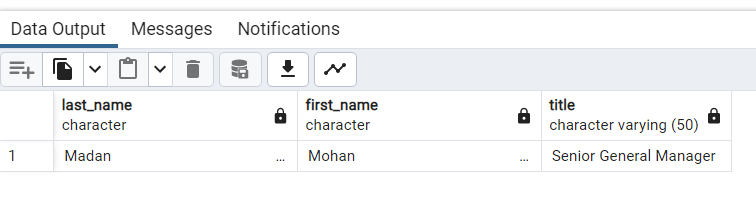
**MUSIC STORE SQL QUERIES**

Q1: Who is the senior most employee based on job title?

Select last\_name, first\_name, title from employee

order by levels desc

limit 1

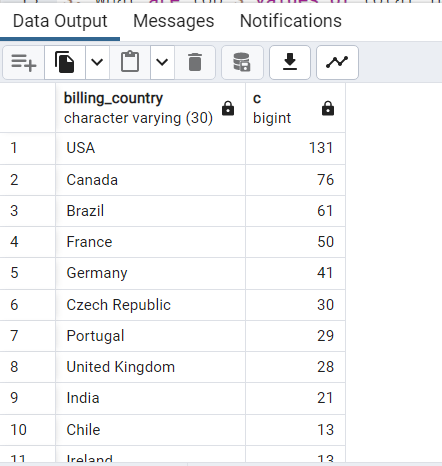


2. Which countries have the most Invoices?

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

group by billing\_country

order by c desc

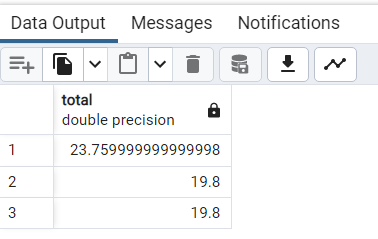


3. What are top 3 values of total invoice?

select total from invoice

order by total desc

limit 3



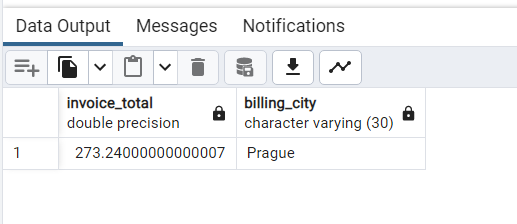
4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals.

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

group by billing\_city

order by invoice\_total 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 customer.customer\_id, first\_name, last\_name, SUM(total) AS total\_spending

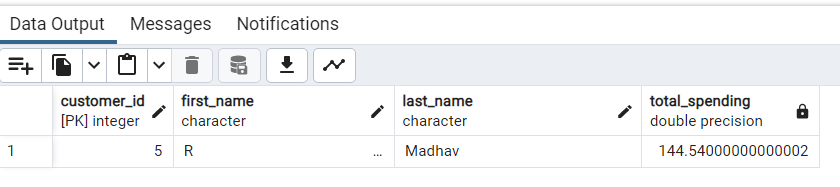
FROM customer

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

GROUP BY customer.customer\_id

ORDER BY total\_spending 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 email,first\_name, last\_name

FROM customer

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

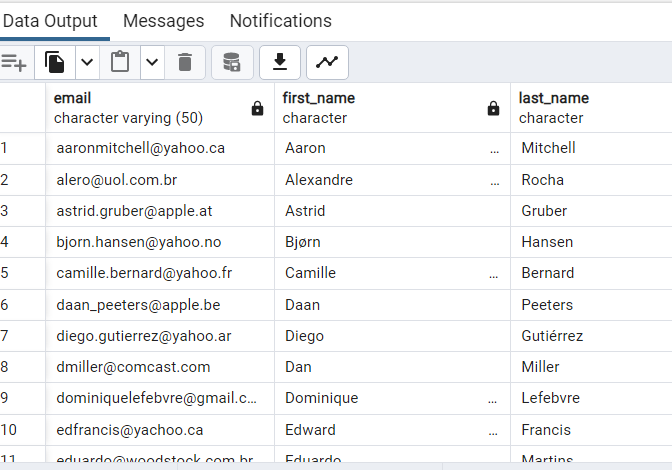
JOIN invoice\_line ON invoice\_line.invoice\_id = invoice.invoice\_id

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

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

WHERE genre.name LIKE 'Rock'

ORDER BY email;



7. 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 ar.name, count(ge.name) as total\_track from artist as ar

join album as al

on ar.artist\_id = al.artist\_id

join track as tr

on tr.album\_id = al.album\_id

join genre as ge

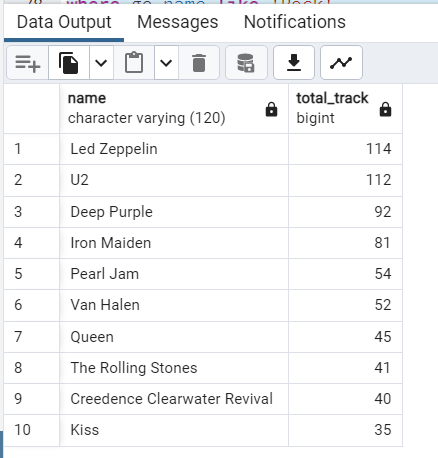
on ge.genre\_id = tr.genre\_id

where ge.name like 'Rock'

group by ar.name

order by total\_track 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, milliseconds from track

where milliseconds >

(select avg(milliseconds) as avg\_length from track)

order by milliseconds desc

