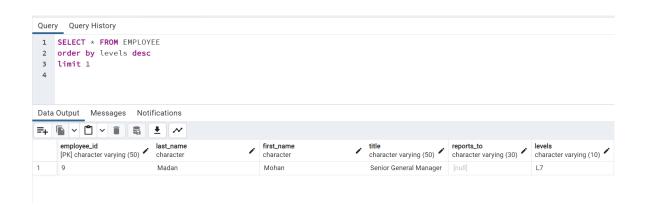
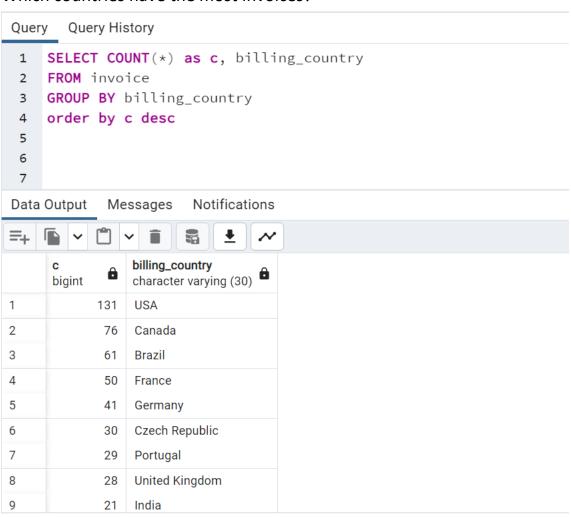
SQL PROJECT- MUSIC STORE DATA ANALYSIS

Questions

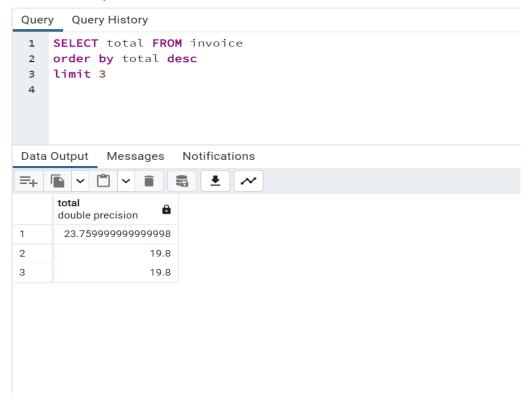
1. Who is the senior most employee based on job title?



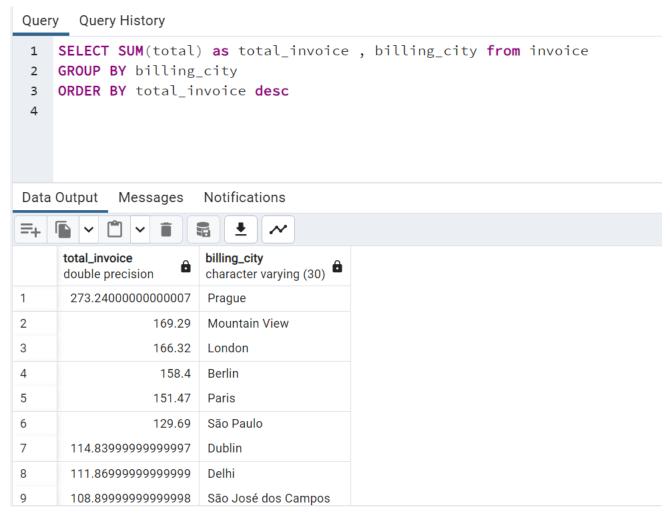
2. Which countries have the most Invoices?



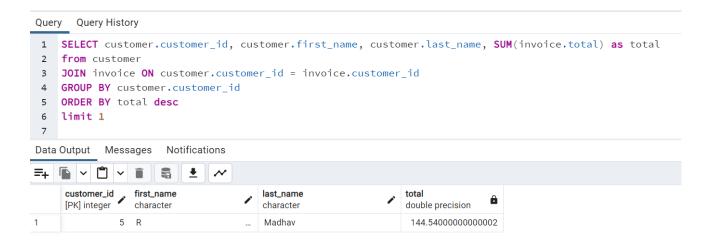
3. What are top 3 values of total invoice?



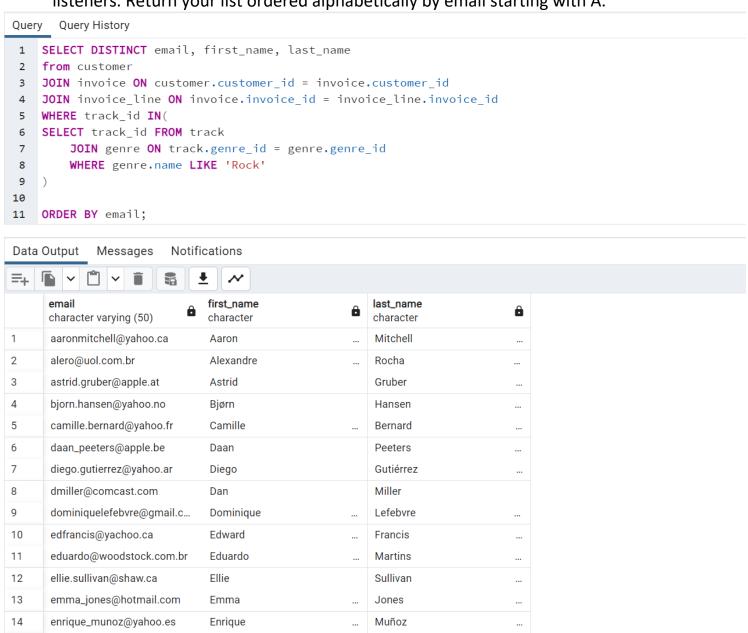
4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals.



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.



6. Write a 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.



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.

Ramos

Harris

15

16

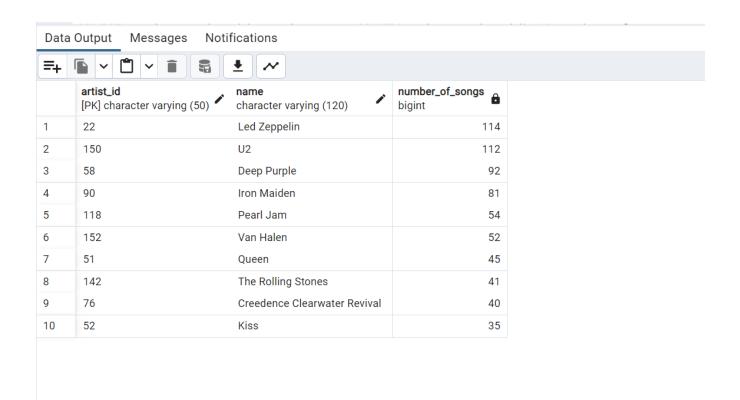
fernadaramos4@uol.com.br

fharris@google.com

Fernanda

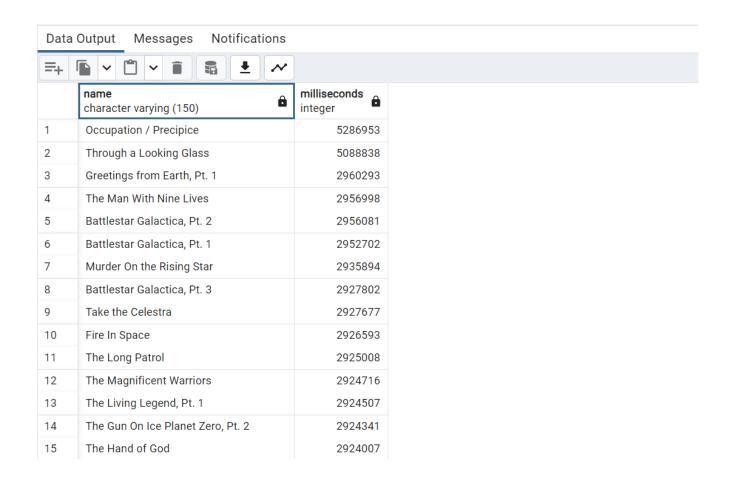
Frank

```
Query
      Query History
1 SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS number_of_songs
    JOIN album ON album.album_id = track.album_id
3
   JOIN artist ON artist.artist_id = album.artist_id
5
   JOIN genre ON genre.genre_id = track.genre_id
   WHERE genre.name LIKE 'Rock'
7
   GROUP BY artist artist id
   ORDER BY number_of_songs DESC
8
9
   LIMIT 10;
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.

```
Query Query History
    SELECT name, milliseconds
 1
 2
    FROM track
 3
    WHERE miliseconds > (
        SELECT AVG(milliseconds) AS avg_track_length
 4
 5
        FROM track)
   ORDER BY milliseconds DESC;
 6
 7
   SELECT name, milliseconds
 8
    FROM track
 9
    WHERE milliseconds > 393599
10
    ORDER BY milliseconds DESC;
11
12
```



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

```
Query
      Query History
    --CTF
1
2
    WITH best_selling_artist AS (
3
        SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
4
        SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
5
        FROM invoice_line
        JOIN track ON track.track_id = invoice_line.track_id
6
        JOIN album ON album.album_id = track.album_id
7
        JOIN artist ON artist.artist_id = album.artist_id
8
        GROUP BY 1
9
10
        ORDER BY 3 DESC
11
        LIMIT 1
12
13
   SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
14
   SUM(il.unit_price*il.quantity) AS amount_spent
15
   FROM invoice i
16
17
    JOIN customer c ON c.customer_id = i.customer_id
   JOIN invoice_line il ON il.invoice_id = i.invoice_id
19
    JOIN track t ON t.track_id = il.track_id
20
    JOIN album alb ON alb.album id = t.album id
   JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
21
22 GROUP BY 1,2,3,4
23 ORDER BY 5 DESC;
```

Data	Output Messa	ages Notifications				
	customer_id integer	first_name character	last_name character	â	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly		Queen	27.719999999999985
2	38	Niklas	Schröder		Queen	18.81
3	3	François	Tremblay		Queen	17.82
4	34	João	Fernandes		Queen	16.8300000000000002
5	53	Phil	Hughes		Queen	11.88
6	41	Marc	Dubois		Queen	11.88
7	47	Lucas	Mancini	***	Queen	10.89
8	33	Ellie	Sullivan		Queen	10.89
9	20	Dan	Miller		Queen	3.96
10	5	R	Madhav		Queen	3.96
11	23	John	Gordon		Queen	2.969999999999998
12	54	Steve	Murray		Queen	2.969999999999998
13	31	Martha	Silk		Queen	2.969999999999998
14	16	Frank	Harris		Queen	1.98
15	17	Jack	Smith		Queen	1.98