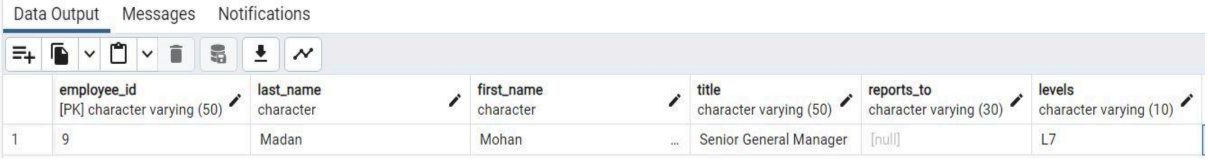
Music Store Data Analysis

By- Vinayak Goyal (goyalvinayak)

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

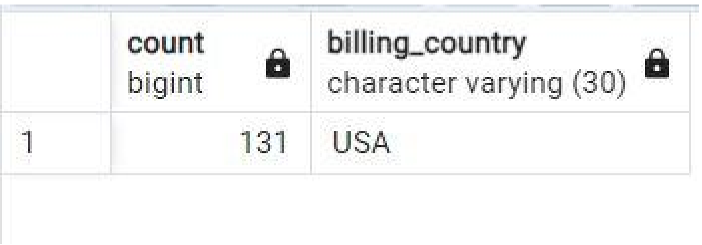
SELECT * FROM employee
ORDER BY levels DESC
 LIMIT 1;



	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (10)
1	9	Madan	Mohan	Senior General Manager	[null]	L7

Ques-2 Which countries have the most Invoices?

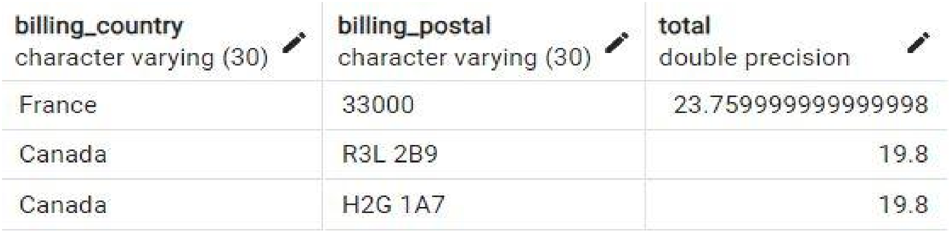
SELECT COUNT(*), billing_country
FROM invoice
GROUP BY billing_country
ORDER BY COUNT(*) DESC
LIMIT 1;



	count bigint	billing_country character varying (30)
1	131	USA

Ques-3 What are the top 3 values of total invoice?

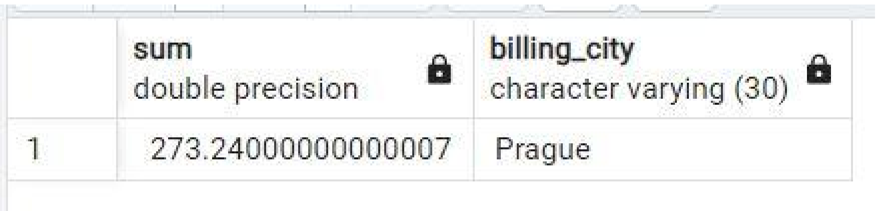
SELECT * FROM invoice
ORDER BY total DESC
LIMIT 3;



billing_country character varying (30)	billing_postal character varying (30)	total double precision
France	33000	23.759999999999998
Canada	R3L 2B9	19.8
Canada	H2G 1A7	19.8

Ques-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.

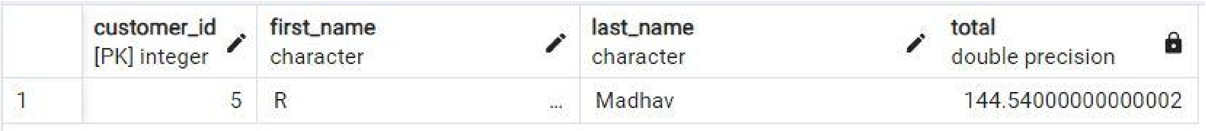
```
SELECT SUM(total), billing_city
FROM invoice
GROUP BY billing_city
ORDER BY SUM(total) DESC
LIMIT 1;
```



	sum double precision	billing_city character varying (30)
1	273.24000000000007	Prague

Ques-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.

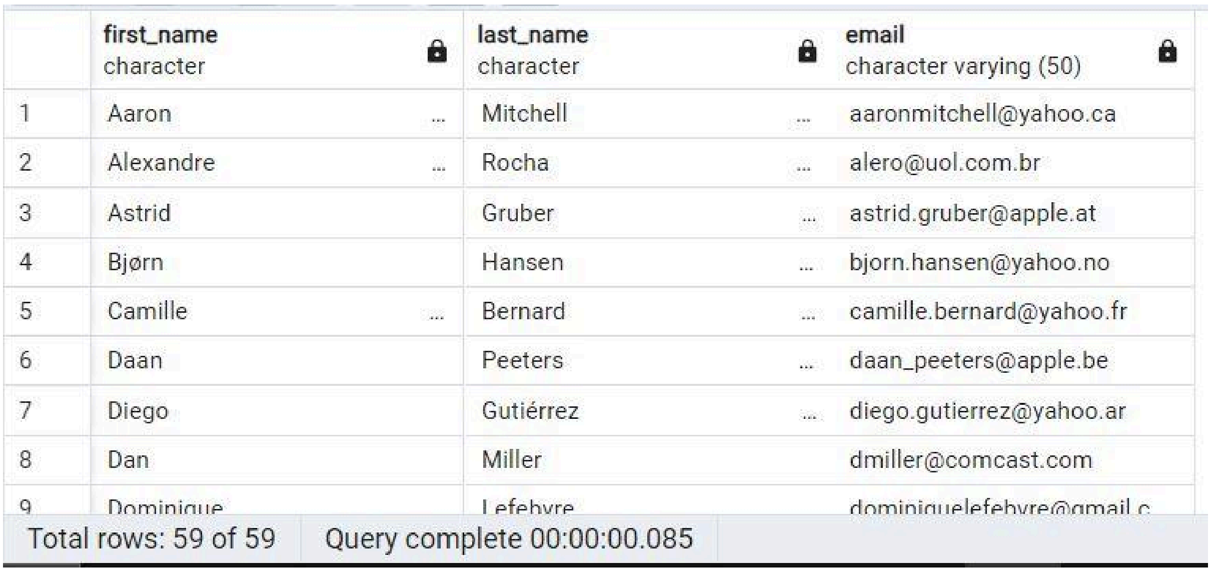
```
SELECT customer.customer_id, customer.first_name, customer.last_name,
SUM(invoice.total) as total
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total DESC
LIMIT 1;
```



	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R	Madhav	144.54000000000002

Ques-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.

```
SELECT DISTINCT first_name, last_name, email
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;
```



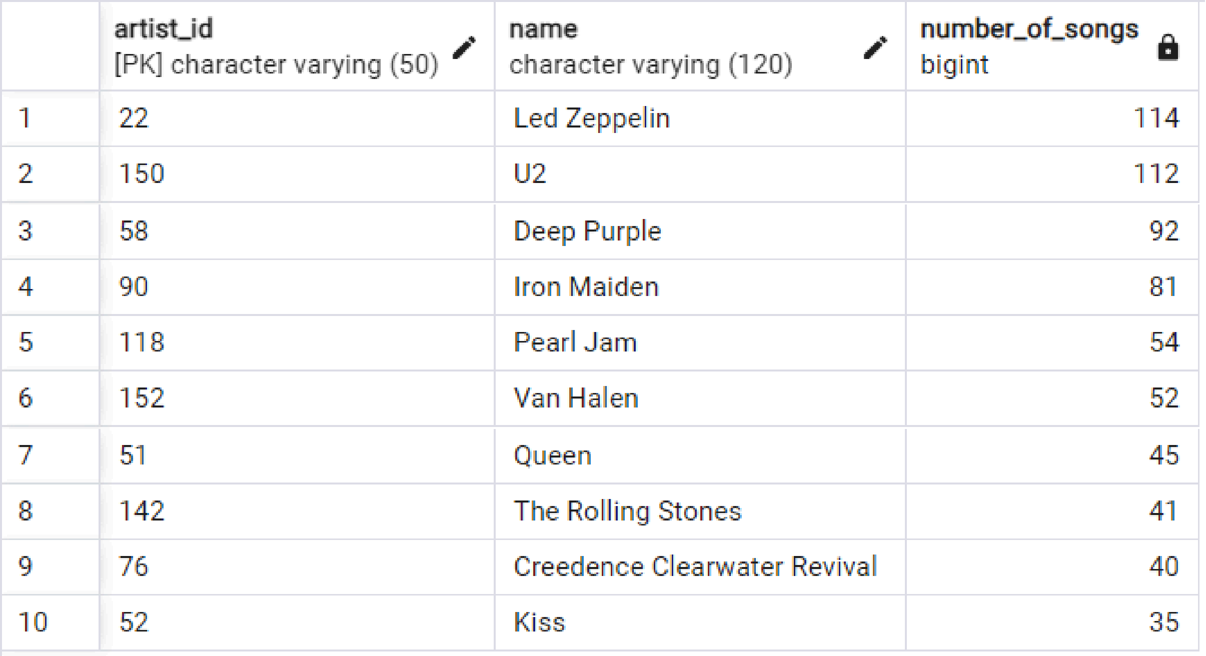
	first_name character	last_name character	email character varying (50)
1	Aaron	Mitchell	aaronmitchell@yahoo.ca
2	Alexandre	Rocha	alero@uol.com.br
3	Astrid	Gruber	astrid.gruber@apple.at
4	Bjørn	Hansen	bjorn.hansen@yahoo.no
5	Camille	Bernard	camille.bernard@yahoo.fr
6	Daan	Peeters	daan_peeters@apple.be
7	Diego	Gutiérrez	diego.gutierrez@yahoo.ar
8	Dan	Miller	dmiller@comcast.com
9	Dominique	Lefebvre	dominiquelefebvre@gmail.c

Total rows: 59 of 59   Query complete 00:00:00.085

Ques-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 artist.artist_id, artist.name, COUNT(artist.artist_id) AS
number_of_songs
FROM track
JOIN album ON track.album_id = album.album_id
JOIN artist ON album.artist_id = artist.artist_id
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name = 'Rock'
```

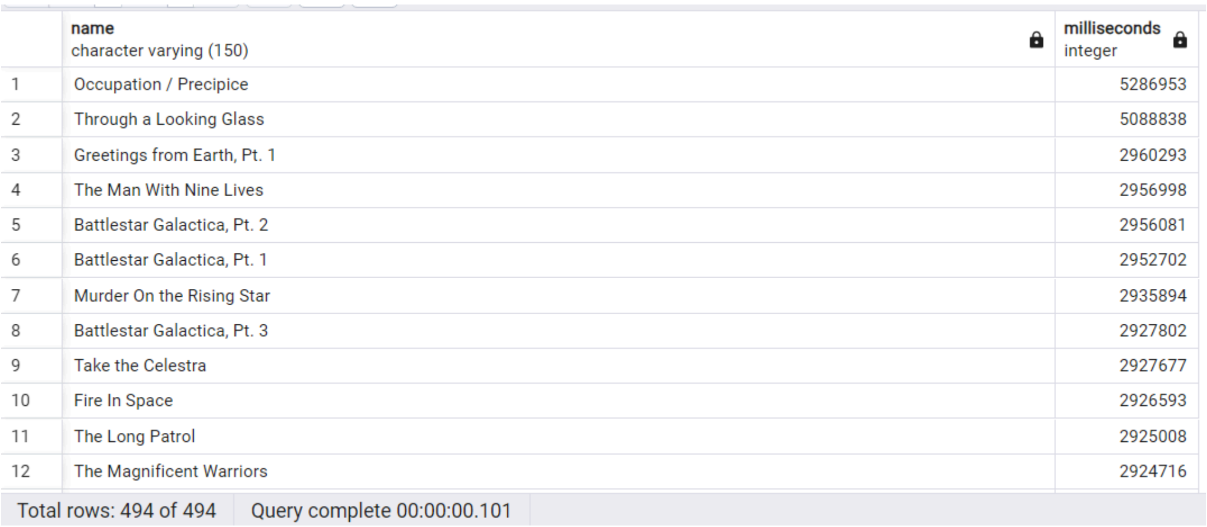
```
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 10;
```



	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

Ques-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)
        FROM track
)
ORDER BY milliseconds DESC;
```

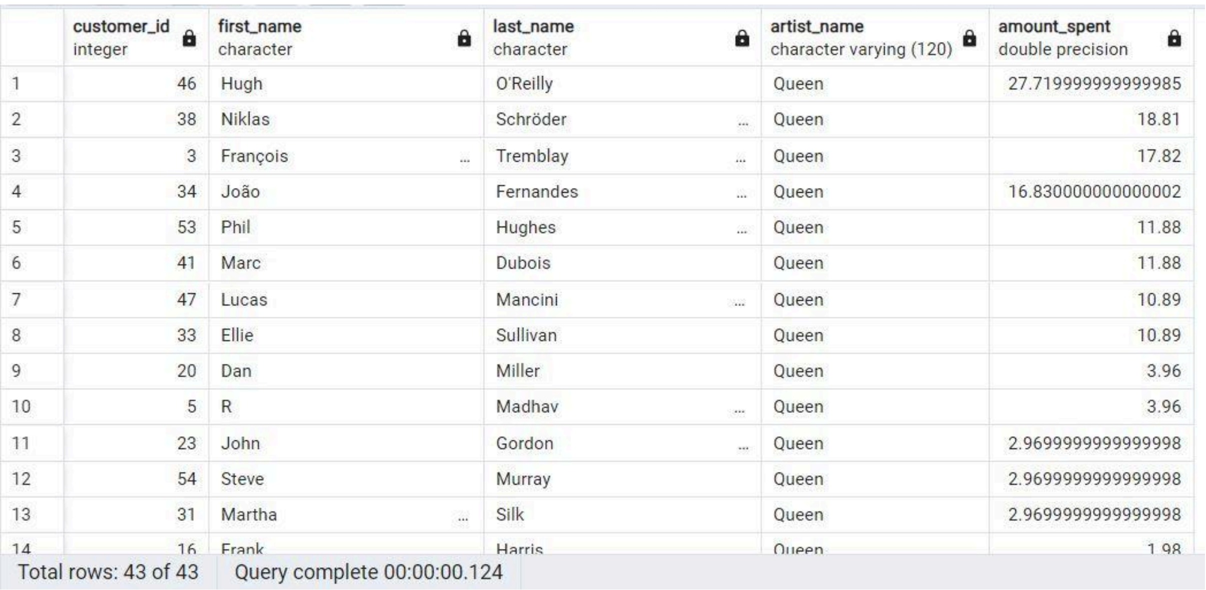


	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716

Total rows: 494 of 494   Query complete 00:00:00.101

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

```
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;
```



	customer_id integer	first_name character	last_name character	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	16.830000000000002
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	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98

Total rows: 43 of 43 Query complete 00:00:00.124