Q1. Who is the Senior Most Employee based JOB TITLE?

```sql
--Q1. who is the senior most employee based job title

SELECT title, last_name, first_name
FROM employee
ORDER BY levels DESC
LIMIT 1
```

Data Output | Messages | Notifications

	title character varying (50)	last_name character (50)	first_name character (50)
1	Senior General Manag...	Madan	Mohan

Q2. Which countries have the most invoice?

Query | Query History

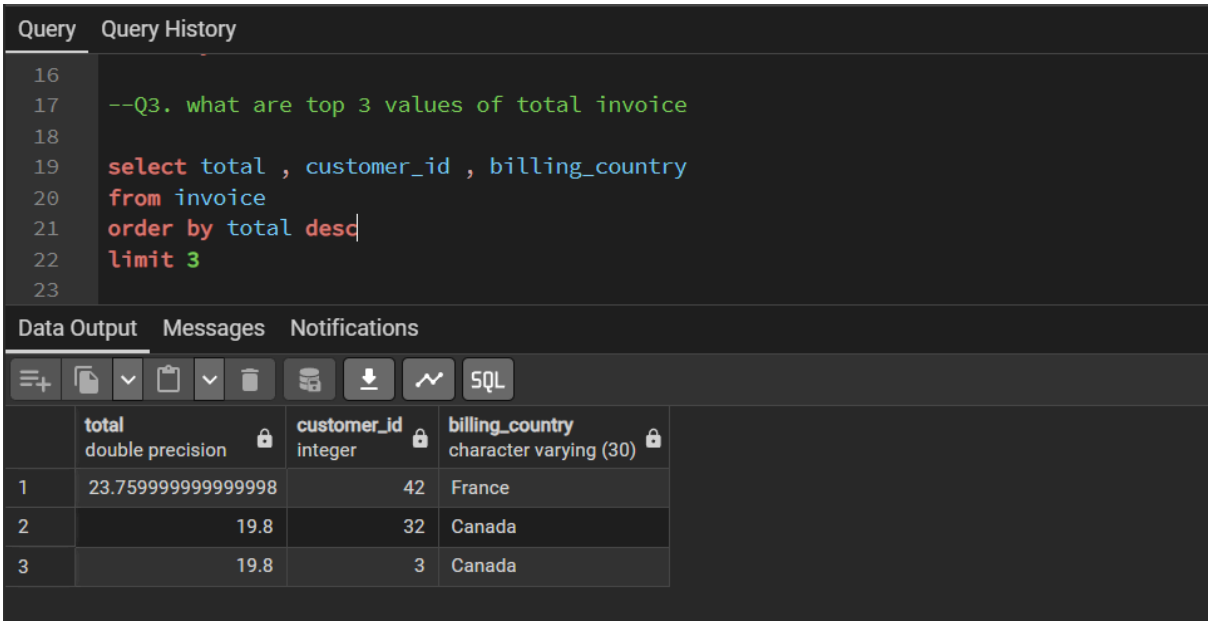
```sql
--Q2. which countries have the most invoices
select * from invoice

select billing_country , count(*) as c
from invoice
group by billing_country
order by c desc
```

Data Output | Messages | Notifications

	billing_country character varying (30)	c bigint
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
7	Portugal	29
8	United Kingdom	28
9	India	21
10	Chile	13
11	Ireland	13
12	Spain	11
13	Finland	11
14	Australia	10

Q3. What are top 3 values of total invoice?

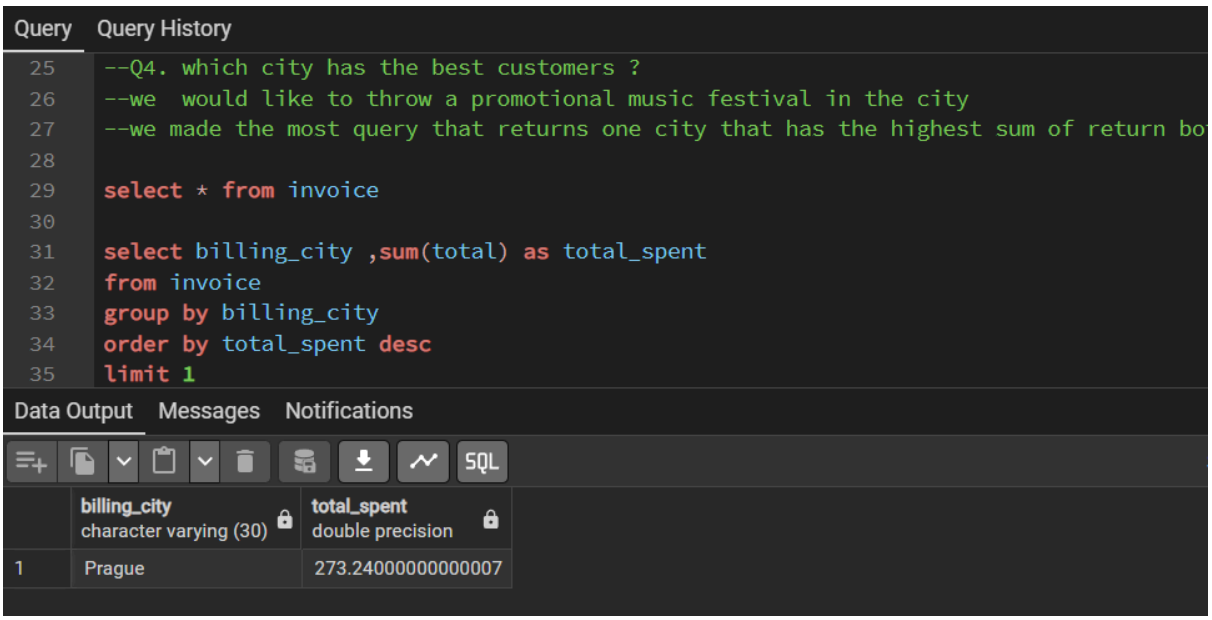


```
Query  Query History
16
17  --Q3. what are top 3 values of total invoice
18
19  select total , customer_id , billing_country
20  from invoice
21  order by total desc
22  limit 3
23
```

Data Output | Messages | Notifications

	total double precision	customer_id integer	billing_country character varying (30)
1	23.759999999999998	42	France
2	19.8	32	Canada
3	19.8	3	Canada

Q4. Which city has the best customers ? we would like to throw a promotional music festival in the city we made the most query that returns one city that has the highest sum of return both city name & sum of all invoices totals ?



```
Query  Query History
25  --Q4. which city has the best customers ?
26  --we  would like to throw a promotional music festival in the city
27  --we made the most query that returns one city that has the highest sum of return bo
28
29  select * from invoice
30
31  select billing_city ,sum(total) as total_spent
32  from invoice
33  group by billing_city
34  order by total_spent desc
35  limit 1
```

Data Output | Messages | Notifications

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

Q5. Who is the best customer ? the customer who has money will be declared the best customer. write a query for the person who has spent the most money?

Query Query History

```
37  --Q5. who is the best customer ?
38  --the customer who has money will be declared the best customer.
39  --write a query the person who has spent the most money?
40  select * from customer
41  select * from invoice
42
43  select customer.customer_id , first_name , last_name , sum(total) as number_of_spent
44  from customer
45  join invoice on customer.customer_id = invoice. invoice_id
46  group by customer.customer_id
47  order by number_of_spent desc
```

Data Output Messages Notifications

SQL Showin

	customer_id [PK] integer	first_name character (50)	last_name character (50)	number_of_spent double precision
1	31	Martha	Silk	19.8
2	54	Steve	Murray	17.82
3	56	Diego	Gutiérrez	16.83
4	5	R	Madhav	16.83
5	1	Luís	Gonçalves	15.84
6	32	Aaron	Mitchell	12.87
7	24	Frank	Ralston	11.879999999999999
8	44	Terhi	Hämäläinen	11.879999999999999
9	52	Emma	Jones	11.879999999999999
10	59	Puja	Srivastava	10.89
11	17	Jack	Smith	10.89

Q6. 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 A ?

Query Query History

```
50  --Q6. Write query to return the email, first name, last name ,
51  --& genre of all rock music listeners .
52  --return your list ordered alphabetically by email starting A
53  select * from customer
54
55  select distinct first_name, last_name , email, genre.name as Name
56  from customer
57  join invoice on  invoice.customer_id= customer.customer_id
58  join invoice_line on invoice_line.invoice_id=invoice.invoice_id
59  join track on track.track_id = invoice_line.track_id
60  join genre on genre.genre_id = track.genre_id
61      where genre.name like 'Rock'
62      order by email asc
```

Data Output Messages Notifications

SQL

	first_name character (50)	last_name character (50)	email character varying (50)	name character varying (120)
1	Aaron	Mitchell	aaronmitchell@yahoo.ca	Rock
2	Alexandre	Rocha	alero@uol.com.br	Rock
3	Astrid	Gruber	astrid.gruber@apple.at	Rock
4	Bjørn	Hansen	bjorn.hansen@yahoo.no	Rock
5	Camille	Bernard	camille.bernard@yahoo.fr	Rock
6	Daan	Peeters	daan_peeters@apple.be	Rock
7	Diego	Gutiérrez	diego.gutierrez@yahoo.ar	Rock
8	Dan	Miller	dmiller@comcast.com	Rock
9	Dominique	Lefebvre	dominiquelefebvre@gmail....	Rock
10	Edward	Francis	edfrancis@yachoo.ca	Rock

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.

```
--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 * from artist
select * from track
select artist.artist_id , artist.name , count(artist.artist_id) as total_track
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 total_track desc
limit 10
```

Data Output | Messages | Notifications

SQL | Showing r

	artist_id [PK] character varying (50)	name character varying (120)	total_track 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 Revi...	40
10	52	Kiss	35

Q8. Return all track names that have song letter the average song length longer than the average song length. Return the name and milliseconds for each track. order by the song length with longest songs listed first.

Query | Query History

```
--Q8. Return all track names that have song letter the average song length longer than the average song length.
--return the name and milliseconds for each track .
--order by the song length with longest songs listed first.
select * from track

select name,milliseconds
from track
where milliseconds > (
    select avg(milliseconds) as avg_track_length
    from track )
order by milliseconds desc;
```

Data Output | Messages | Notifications

SQL | Showing rows: 1 to 494 | Page

	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

--Q9. 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

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

select customer.customer_id , customer.first_name , customer.last_name , billing_country,
artist.name as artist_name, sum(total) as total_spent
from customer
join invoice on customer.customer_id = invoice.invoice_id
join invoice_line on invoice.invoice_id = invoice_line.invoice_id
join track on invoice_line.invoice_id = track.track_id
join album on track.album_id = album.album_id
join artist on album.album_id = artist.artist_id
group by customer.customer_id , customer.first_name , customer.last_name , billing_country ,artist.name
order by total_spent desc
```

Data Output Messages Notifications

Showing rows: 1 to 59

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	artist_name character varying (120)	total_spent double precision
1	31	Martha	Silk	Canada	Alice In Chains	396.0000000000001
2	54	Steve	Murray	USA	Apocalyptica	320.75999999999993
3	5	R	Madhav	USA	Aerosmith	286.10999999999984
4	56	Diego	Gutiérrez	Portugal	Apocalyptica	286.10999999999984
5	1	Luís	Gonçalves	USA	AC/DC	253.44000000000003
6	32	Aaron	Mitchell	Canada	Alice In Chains	167.31000000000003
7	24	Frank	Ralston	Norway	Alice In Chains	142.55999999999997
8	52	Emma	Jones	USA	Apocalyptica	142.55999999999997
9	44	Terhi	Hämäläinen	USA	Antônio Carlos Jobim	142.55999999999997
10	59	Puja	Srivastava	Germany	Apocalyptica	119.79

--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.

Query Query History

```
--Q10. We want to find out the most popular music genre for each country.
--we determine the most popular genre as the genre with 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 purchase ,
        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_line.invoice_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
```

Data Output Messages Notifications

Showing rows: 1 to 24 Page N

	purchase bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Rock	1	1
2	7	Australia	Rock	1	1
3	6	Austria	Rock	1	1
4	6	Belgium	Rock	1	1
5	28	Brazil	Rock	1	1

Q11. Write a query that determines the customer that has on music for each country. write a query that returns that with top customer and how much they spent. for company the top amount spent is shared , provide all customers with amount.

Query Query History

```
129  --Q11. Write a query that determines the customer that has on music for each country.
130  --write a query that returns that with top customer and how much they spent.
131  --for company the top amount spent is shared , provide all customers with amount.
132
133  select * from customer
134  select * from invoice
135
136  with customer_with_country as (
137      select customer.customer_id,first_name ,last_name , billing_country , sum(total) as total_spending,
138      row_number() over(partition by billing_country order by sum(total)desc) as RowNo
139      from invoice
140      join customer on customer.customer_id = invoice.customer_id
141      group by 1,2,3,4
142      order by 4 asc , 5 desc
143  )select * from customer_with_country where RowNo <= 1
```

Data Output Messages Notifications

SQL Showing rows: 1 to 24

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1