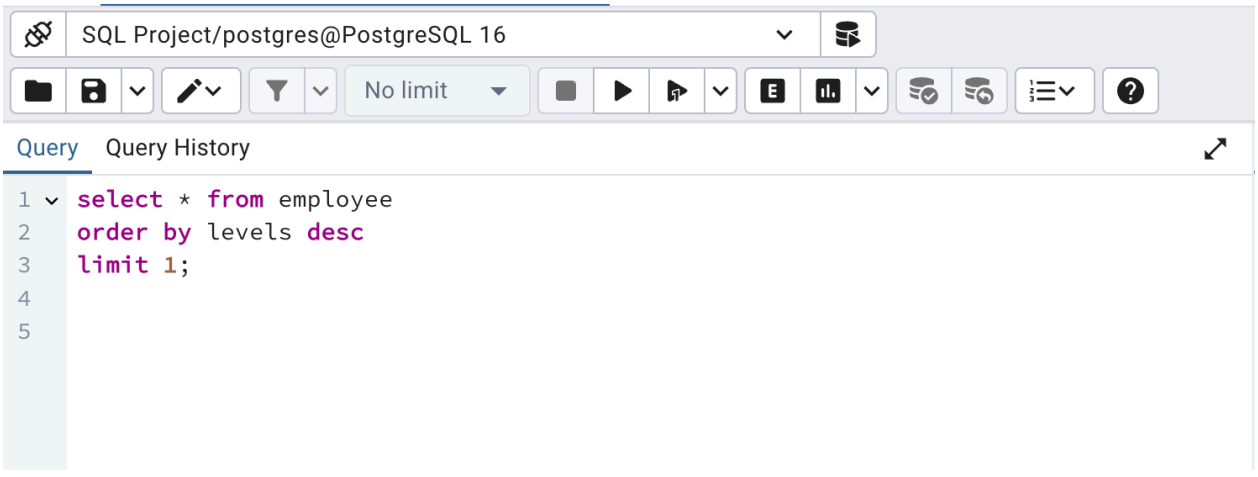
Music Store Data Analysis Using SQL

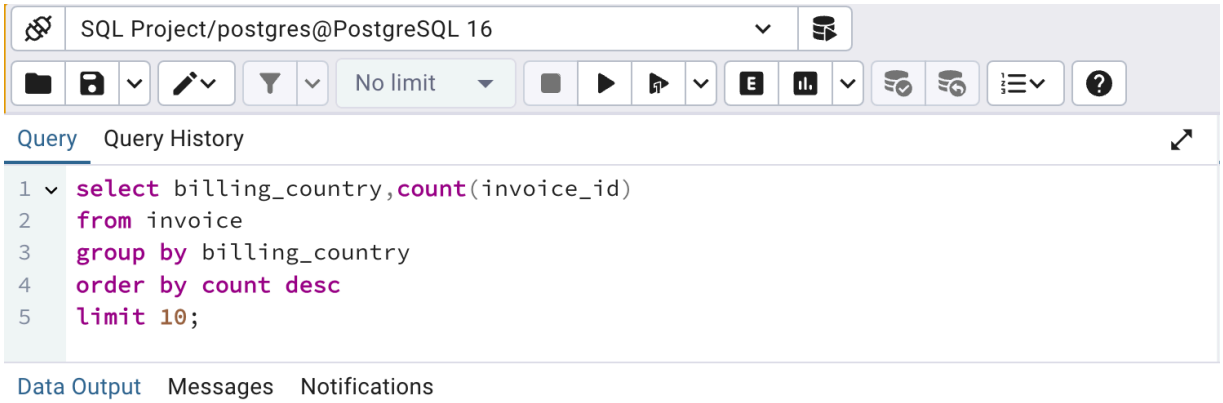
Query-1 : Who is the senior most employee based on job title?



Data Output | Messages | Notifications

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

Query- 2 : Top 10 countries have the most Invoices?



Data Output | Messages | Notifications

	billing_country character varying (30)	count 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

Query 3 : What are top 3 values of total invoice?

Query | Query History

```
1  select billing_country, total from invoice
2  order by total desc
3  limit 3
```

Data Output | Messages | Notifications

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

Query 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

Query | Query History

```
1  select sum(total) as invoice_total,billing_city from invoice
2  group by billing_city
3  order by invoice_total desc
4  limit 5;
5
```

Data Output | Messages | Notifications

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris

Query 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.

Query | Query History

```
select customer.customer_id,first_name, last_name, sum(total) as total from customer
join invoice on customer.customer_id = invoice.customer_id
group by customer.customer_id,first_name,last_name
order by total desc
limit 3;
```

Data Output | Messages | Notifications

	customer_id [PK] integer	first_name character (50)	last_name character (50)	total double precision
1	5	R	Madhav	144.54000000000002
2	6	Helena	Holý	128.7
3	46	Hugh	O'Reilly	114.83999999999997

Query 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.

Query | Query History

```
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;
```

Data Output | Messages | Notifications

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

Query 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.

Query | Query History

```sql
select distinct artist.name,count(track.name) as rock_band
from artist join album on artist.artist_id = album.artist_id
join track on album.album_id = track.album_id
join genre on track.genre_id = genre.genre_id
where genre.name = 'Rock'
group by artist.name
order by rock_band desc
limit 10;
```

Data Output | Messages | Notifications

	name character varying (120)	rock_band bigint
1	Led Zeppelin	114
2	U2	112
3	Deep Purple	92
4	Iron Maiden	81
5	Pearl Jam	54
6	Van Halen	52
7	Queen	45
8	The Rolling Stones	41
9	Creedence Clearwater Revival	40
10	Kiss	35

Query 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

```
1  select distinct track.name , milliseconds from track
2  where milliseconds > (
3  select avg(milliseconds) from track
4  )
5  order by milliseconds desc
6  limit 10
```

Data Output | Messages | Notifications

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt...	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

Query 9 :

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

SQL Project/postgres@PostgreSQL 16

No limit

Query | Query History

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

Data Output | Messages | Notifications

SQL

	customer_id integer	first_name character (50)	last_name character (50)	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

Query 10 :

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

```
with popular_genre as (
  select distinct genre.genre_id, genre.name,customer.country as country,
  count(invoice_line.quantity) as purchase,ROW_NUMBER() OVER(PARTITION BY customer.country
  ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
  from invoice_line
  join invoice on invoice_line.invoice_id = invoice.invoice_id
  join customer on invoice.customer_id = customer.customer_id
  join track on invoice_line.track_id = track.track_id
  join genre on track.genre_id = genre.genre_id
  group by 1,2,3
  order by 4 desc, 3 asc
)
select*from popular_genre where RowNo <=1
limit 10
```

Data Output | Messages | Notifications

	genre_id character varying (50)	name character varying (120)	country character varying (50)	purchase bigint	rowno bigint
1	1	Rock	USA	561	1
2	1	Rock	Canada	333	1
3	1	Rock	France	211	1
4	1	Rock	Brazil	205	1
5	1	Rock	Germany	194	1
6	1	Rock	United Kingdom	166	1
7	1	Rock	Czech Republic	143	1
8	1	Rock	Portugal	108	1
9	1	Rock	India	102	1
10	1	Rock	Ireland	72	1

Query 11 :

Write a query that determines the customer that has spent the most 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.

Query | Query History

```sql
WITH customer_with_country AS (
    SELECT
        customer.country AS billing_country,
        customer.first_name,
        SUM(invoice.total) AS Total_Spending
    FROM
        customer JOIN invoice
    ON
    customer.customer_id = invoice.customer_id
    GROUP BY
        customer.country, customer.first_name
),country_max 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
FROM customer_with_country cc
     JOIN country_max mx
ON  cc.billing_country = mx.billing_country
     AND cc.Total_Spending = mx.max_spending
ORDER BY cc.billing_country
      limit 10;
```

	billing_country character varying (50)	total_spending double precision	first_name character (50)
1	Argentina	39.6	Diego
2	Australia	81.18	Mark
3	Austria	69.3	Astrid
4	Belgium	60.38999999999999	Daan
5	Brazil	108.89999999999998	Luís
6	Canada	99.99	François
7	Chile	97.02000000000001	Luis
8	Czech Republic	144.54000000000002	R
9	Denmark	37.61999999999999	Kara
10	Finland	79.2	Terhi