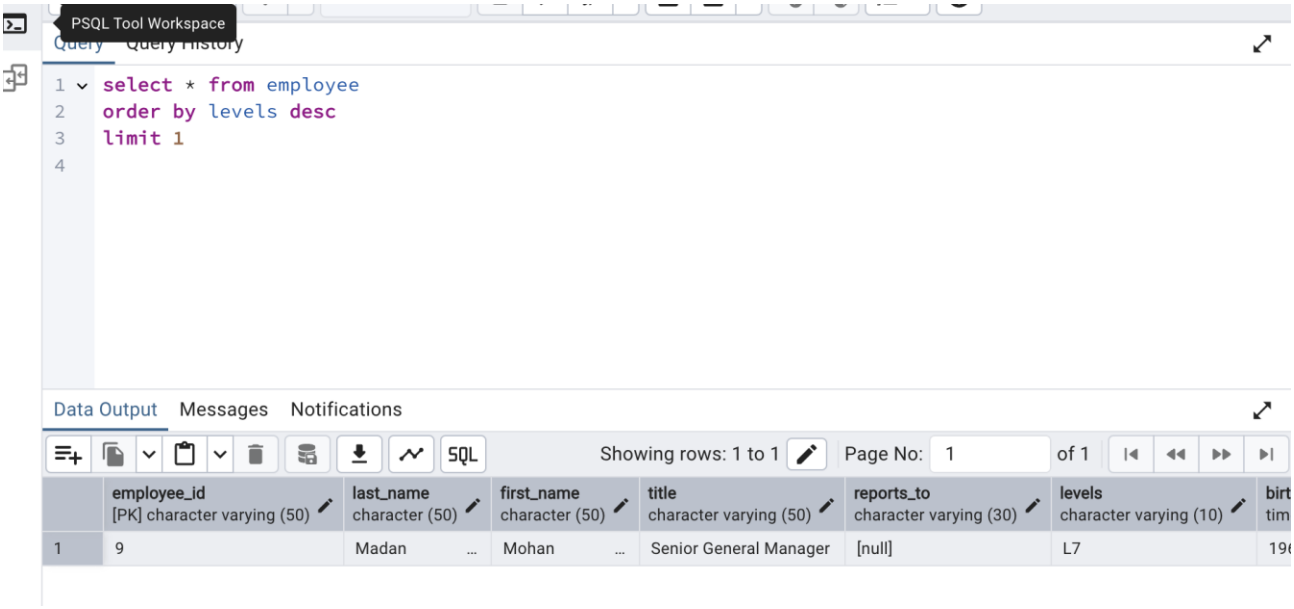
SQL PROJECT- MUSIC STORE DATA ANALYSIS

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



2. Which countries have the most Invoices?

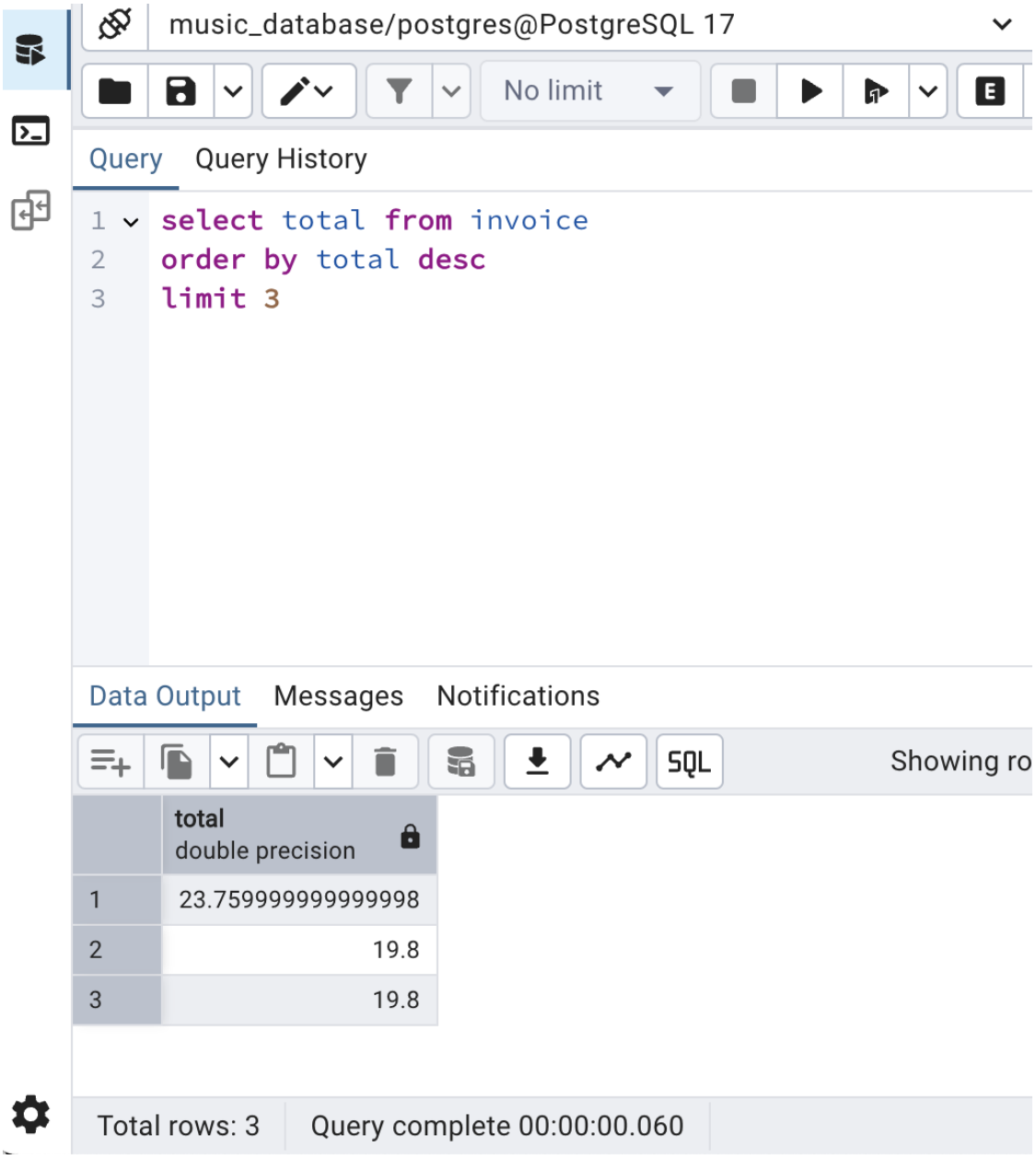
```
select count(*) as c, billing_country
from invoice
group by billing_country
order by c desc
```

Data Output | Messages | Notifications

Showing rows: 1 to 24

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

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.

```sql
select billing_city , sum(total) as invoice_total
from invoice
group by billing_city
order by invoice_total desc
limit 1
```

Data Output | Messages | Notifications

Showing rows:

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

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.

```sql
select c.customer_id , first_name,last_name, sum(total) as most_money
from customer c
join invoice i on c.customer_id=i.customer_id
group by c.customer_id
order by most_money desc
limit 1
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1

	customer_id [PK] integer	first_name character (50)	last_name character (50)	most_money double precision
1	5	R ...	Madhav ...	144.54000000000002

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.

```sql
select 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='Rock'
)
order by email;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1000 | Page No: 1 of 3

	email character varying (50)	first_name character (50)	last_name character (50)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	aaronmitchell@yahoo.ca	Aaron	Mitchell
3	aaronmitchell@yahoo.ca	Aaron	Mitchell
4	aaronmitchell@yahoo.ca	Aaron	Mitchell
5	aaronmitchell@yahoo.ca	Aaron	Mitchell
6	aaronmitchell@yahoo.ca	Aaron	Mitchell
7	aaronmitchell@yahoo.ca	Aaron	Mitchell

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.

```sql
select at.artist_id , at.name ,count(at.artist_id) as track_count
from track t
join album a on a.album_id = t.album_id
join artist at on at.artist_id = a.artist_id
join genre g on g.genre_id = t.genre_id
where g.name like 'Rock'
group by at.artist_id
order by track_count desc
limit 10
```

Data Output | Messages | Notifications

	artist_id [PK] character varying (50)	name character varying (120)	track_count 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

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.

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

Data Output | Messages | Notifications

Showing rows: 1 to 494 | Page No: 1 of 1

	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

Total rows: 494 | Query complete 00:00:00.097

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

```sql
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
```

Data Output | Messages | Notifications

	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
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

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.

```sql
with popular_genre as
(
    select count(invoice_line.quantity) as purchases, customer.country , genre.name, genre.genre_id,
    Row_number() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity)desc) as RowN
    from invoice_line
    join invoice on invoice.invoice_id = invoice_line.invoice_id
    join customer on customer.customer_id=invoice.customer_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 No: 1 of 1

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1