



MUSIC STORE ANALYSIS

SQL PROJECT

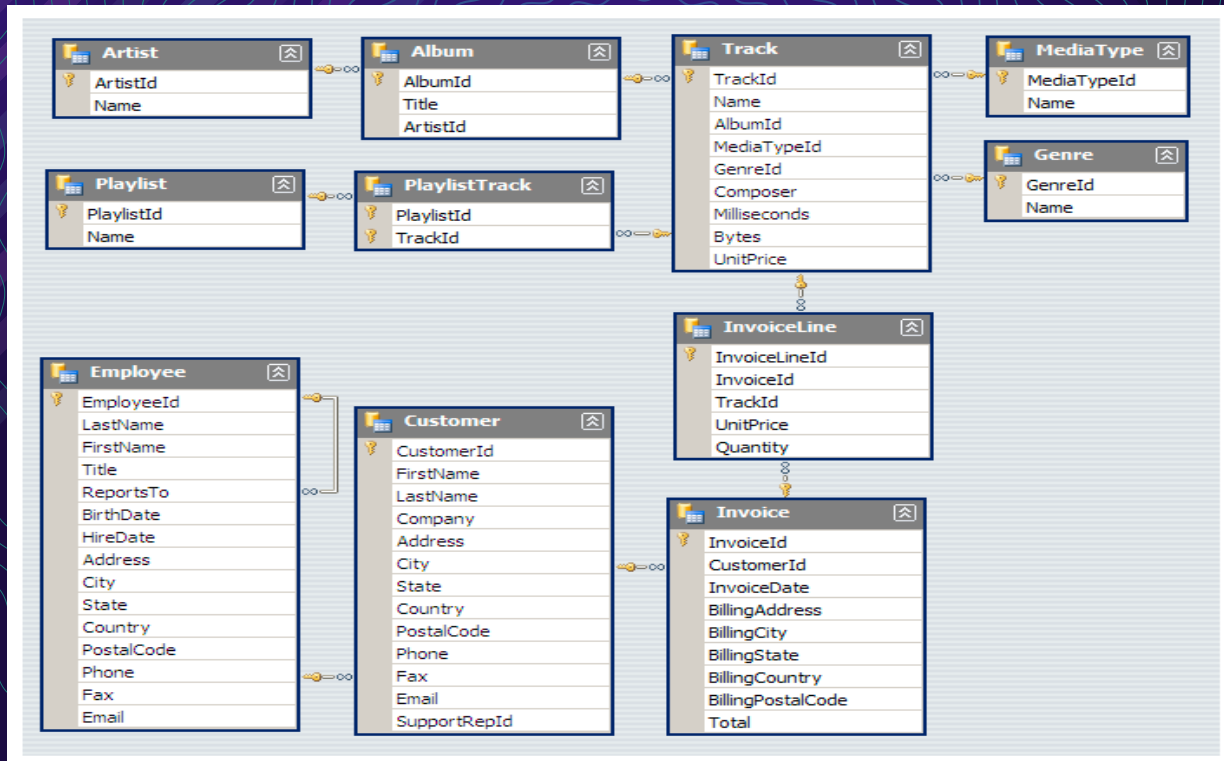
BY ASHUTOSH

OBJECTIVE

- The primary objective of the music store is to achieve sustainable business growth while addressing existing challenges.
- We need to examine the dataset with SQL and help the music store understand its business growth by answering simple questions

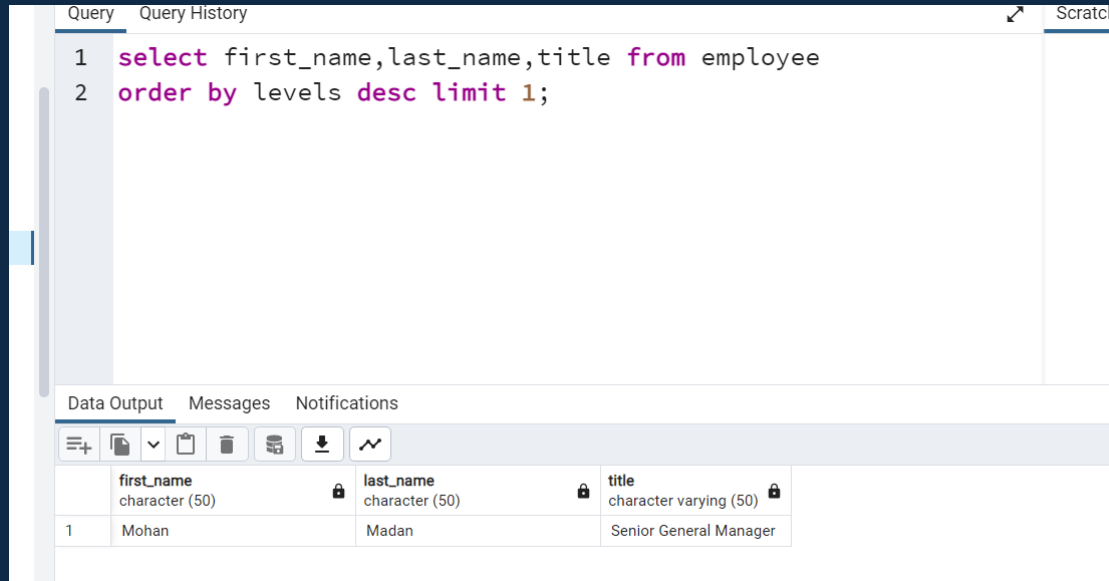


TABLE SCHEMA



SQL Analysis with their Output

Q1. Who is the senior most employee based on job title?



The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 select first_name,last_name,title from employee
2 order by levels desc limit 1;
```

The results pane, titled "Data Output", displays the output of the query in a table format. The table has four columns: "first_name", "last_name", and "title". The data row shows the following values:

| | first_name character (50) | last_name character (50) | title character varying (50) |
|---|------------------------------|-----------------------------|---------------------------------|
| 1 | Mohan | Madan | Senior General Manager |

Q2. Which countries have the most Invoices?

Query

Query History

Scratch Pad

```
1 select billing_country,count(*) as no_of_invoice from invoice
2 group by billing_country order by no_of_invoice desc
```

Data Output

Messages

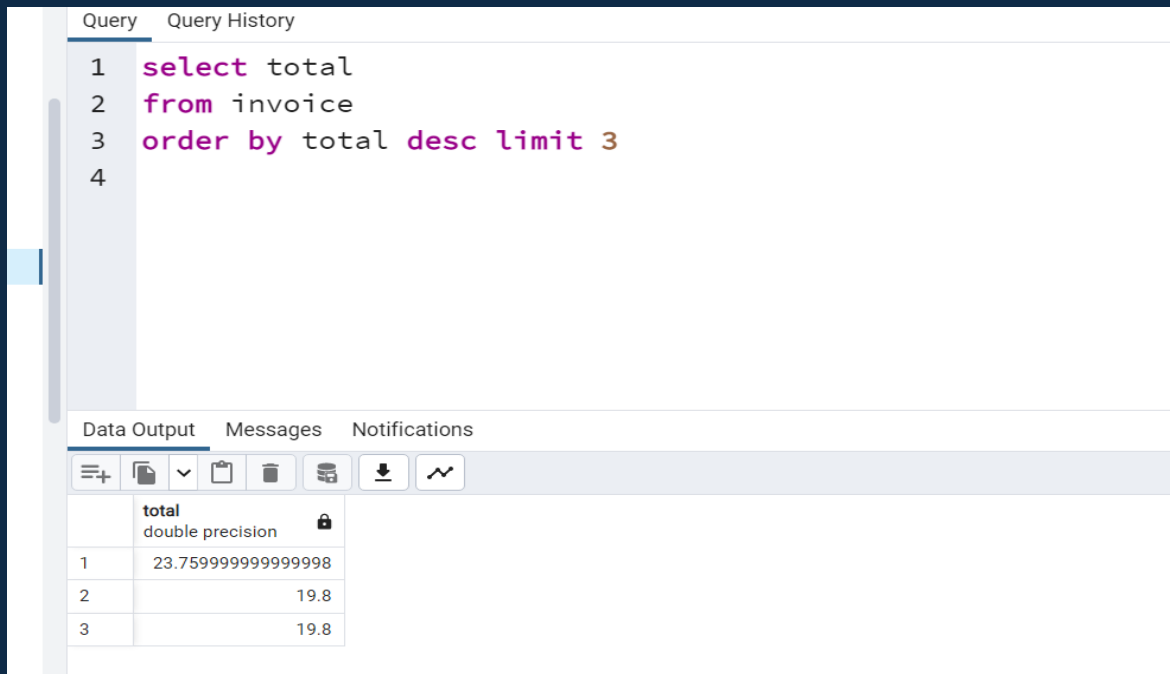
Notifications

| | billing_country character varying (30) | no_of_invoice 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 |

Total rows: 24 of 24

Query complete 00:00:00.094

Q3 What are top 3 values of total invoice?

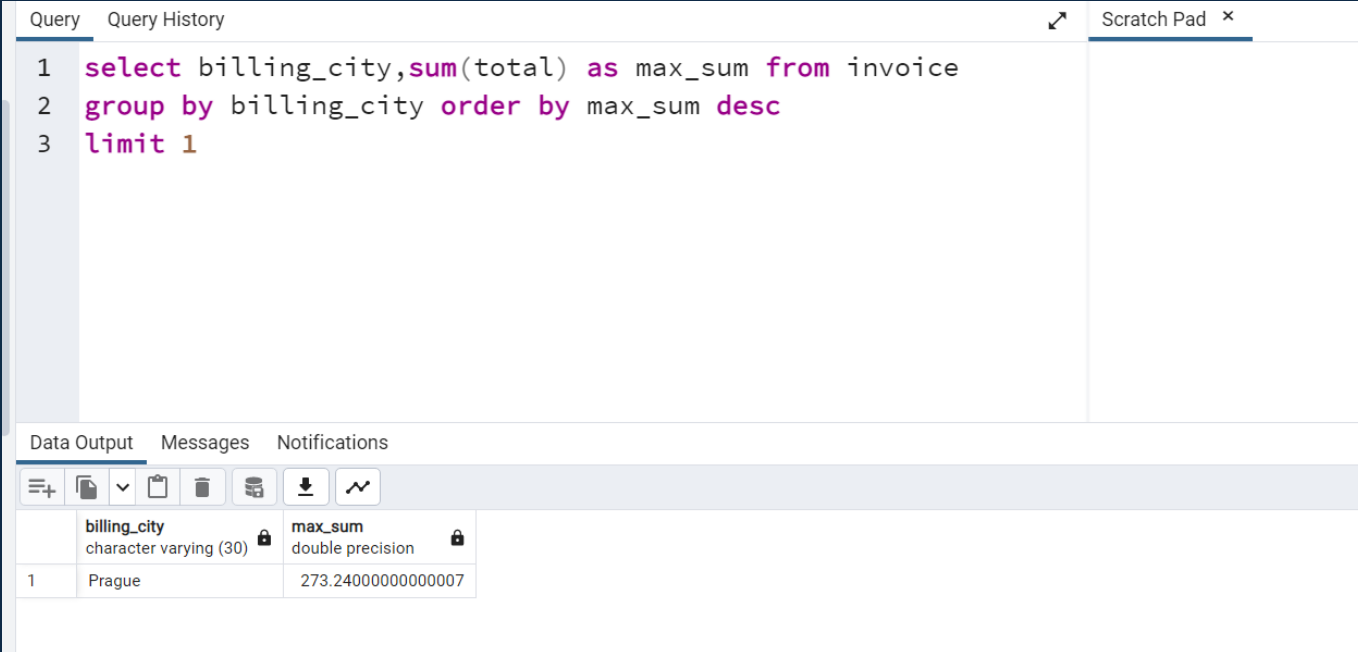


The screenshot shows a SQL query editor with a query that selects the top 3 values of the 'total' column from an 'invoice' table, ordered in descending order. Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table format. The table has two columns: 'total' (double precision) and an unnamed column. The results show three rows: the first row has a total of 23.759999999999998, and the next two rows have a total of 19.8.

```
1 select total
2 from invoice
3 order by total desc limit 3
4
```

| | total double precision | |
|---|---------------------------|------|
| 1 | 23.759999999999998 | |
| 2 | | 19.8 |
| 3 | | 19.8 |

Q4. 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



The screenshot shows a SQL query editor interface. The top bar has tabs for 'Query' and 'Query History', and a 'Scratch Pad' tab on the right. The main area contains a SQL query:

```
1 select billing_city, sum(total) as max_sum from invoice
2 group by billing_city order by max_sum desc
3 limit 1
```

Below the query editor, there is a 'Data Output' tab, 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'billing_city' (character varying (30)) and 'max_sum' (double precision). The table contains one row with the value 'Prague' for the city and '273.240000000000007' for the sum.

| | billing_city character varying (30) | max_sum double precision |
|---|--|-----------------------------|
| 1 | Prague | 273.240000000000007 |

Q5. 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

```
1 select c.customer_id,c.first_name,c.last_name,sum(total) as highest_spent
2 from customer as c inner join invoice as i on c.customer_id=i.customer_id
3 group by c.customer_id order by highest_spent desc limit 1
4
```

Data Output

Messages

Notifications



| | customer_id [PK] integer | first_name character (50) | last_name character (50) | highest_spent double precision |
|---|-----------------------------|------------------------------|-----------------------------|-----------------------------------|
| 1 | 5 | R | Madhav | 144.54000000000002 |

Q6. 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

```
1 select distinct c.email,c.first_name,c.last_name,g.name from customer c
2 join invoice i on c.customer_id=i.customer_id
3 join invoice_line il on i.invoice_id=il.invoice_id
4 join track t on il.track_id=t.track_id
5 join genre g on g.genre_id=t.genre_id where g.name='Rock'
6 order by c.email
```

Data Output

Messages

Notifications

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

Query









Query History

```
1 select artist.artist_id,artist.name,count(artist.artist_id) as number_of_songs
2 from artist join album on artist.artist_id = album.artist_id
3 join track on album.album_id=track.album_id
4 join genre on track.genre_id=genre.genre_id
5 where genre.name='Rock'
6 group by artist.artist_id order by number_of_songs desc limit 10;
7
```

Data Output

Messages

Notifications



| | 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 |

Total rows: 10 of 10

Query complete 00:00:00.135

Q8. 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 name,milliseconds from track where milliseconds >
2 (select avg(milliseconds) from track)
3 order by milliseconds desc
```

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑️

🗄️

⬇️

📈

| | name | milliseconds |
|---|-----------------------------|--------------|
| | character varying (150) | 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 |

Total rows: 494 of 494

Query complete 00:00:00.213

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
1  with best_selling_artist as(
2  select at.artist_id,at.name,sum(il.unit_price*il.quantity) as total_sales
3  from artist as at join album as a on at.artist_id=a.artist_id
4  join track as t on a.album_id=t.album_id
5  join invoice_line as il on t.track_id=il.track_id
6  group by at.artist_id
7  order by total_sales desc
8  limit 1
9  )
10
11 select c.customer_id,c.first_name,c.last_name,bsa.name,
12 sum(il.unit_price*il.quantity) as total_spent
13 from customer as c join invoice as i on c.customer_id=i.customer_id
14 join invoice_line as il on i.invoice_id=il.invoice_id
15 join track as t on il.track_id=t.track_id
16 join album as a on t.album_id=a.album_id
17 join best_selling_artist as bsa on a.artist_id=bsa.artist_id
18 group by c.customer_id,bsa.name
19 order by total_spent desc
20

```

Output:-

| | customer_id integer | first_name character (50) | last_name character (50) | name character varying (120) | total_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 |

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
1  with popular_genre as (
2  select g.name,c.country,count(il.quantity) as purchases,
3  row_number() over(partition by c.country order by count(il.quantity) desc) as Rownum
4  from customer c join invoice i on c.customer_id=i.customer_id
5  join invoice_line il on i.invoice_id=il.invoice_id
6  join track t on il.track_id=t.track_id
7  join genre g on t.genre_id=g.genre_id
8  group by g.name,c.country
9  order by c.country asc,purchases desc
10 )
11
12 select * from popular_genre where Rownum<=1
```

Output:-

| Data Output | | | | | Messages | Notifications |
|----------------------|---------------------------------|-----------------------------------|-----------------------------|------------------|----------|---------------|
| | | | | | | |
| | name character varying (120) | country character varying (50) | purchases bigint | rownum bigint | | |
| 1 | Alternative & Punk | Argentina | 17 | 1 | | |
| 2 | Rock | Australia | 34 | 1 | | |
| 3 | Rock | Austria | 40 | 1 | | |
| 4 | Rock | Belgium | 26 | 1 | | |
| 5 | Rock | Brazil | 205 | 1 | | |
| 6 | Rock | Canada | 333 | 1 | | |
| 7 | Rock | Chile | 61 | 1 | | |
| 8 | Rock | Czech Republic | 143 | 1 | | |
| 9 | Rock | Denmark | 24 | 1 | | |
| 10 | Rock | Finland | 46 | 1 | | |
| Total rows: 24 of 24 | | | Query complete 00:00:00.123 | | | |

Q11. 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

```

1 with popular_genre as (
2 select c.customer_id,c.first_name,c.last_name,i.billing_country,
3 sum(i.total) as total_spending,
4 row_number() over(partition by i.billing_country order by sum(i.total) desc) as Rownum
5 from customer c join invoice i on c.customer_id=i.customer_id
6 group by 1,2,3,4
7 order by 4 asc,5 desc
8 )
9 select * from popular_genre where Rownum<=1

```

Data Output
Messages
Notifications

| | customer_id integer | first_name character (50) | last_name character (50) | billing_country character varying (30) | total_spending double precision | rownum 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 | 60.389999999999999 | 1 |
| 5 | 1 | Luis | Gonçalves | Brazil | 108.899999999999998 | 1 |
| 6 | 3 | François | Tremblay | Canada | 99.99 | 1 |
| 7 | 57 | Luis | Rojas | Chile | 97.020000000000001 | 1 |

Total rows: 24 of 24
Query complete 00:00:00.133
Ln 5, Col 1

INSIGHTS

Employee Distribution: There are fewer employees at the senior level compared to the junior level.

Customer Countries: The majority of customers, approximately 70%, are from European countries.

Top Revenue Source: The maximum total invoice amount comes from Prague city. Additionally, the top two customers who spend the most money also belong to Prague.

Popular Music Genre: Rock is the only music genre that is popular in all countries except Argentina in terms of purchase.

- **Leading Rock Artist:** Led Zeppelin is the artist who has written the most rock music.



THANKS