Music Store Data Analysis Project using SQL

Project Overview

The Music Store Data Analysis Project aims to explore and analyze data from a music store database using SQL queries. This project addresses various problems ranging from identifying top employees and customers to determining popular music genres and high-revenue cities. The analysis is divided into three levels of difficulty: easy, intermediate, and advanced. Each level tackles different aspects of the data to provide valuable insights into the music store's operations and customer preferences.

Dataset Description

The dataset consists of several tables related to the music store's operations, including:

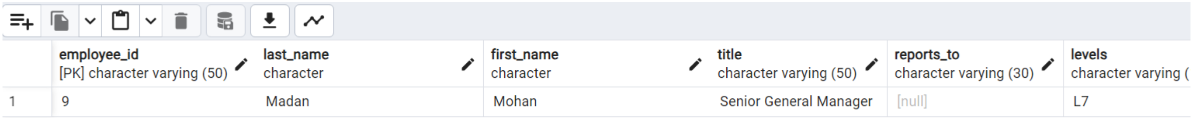
- **employee:** Details about employees, including their job titles and levels.
- **invoice:** Information about invoices, including billing details and total amounts.
- **customer:** Customer information such as names and email addresses.
- **invoice_line:** Details of each item included in an invoice.
- **track:** Information about individual tracks, including genre and album details.
- **album:** Album details including the artist.
- **artist:** Information about artists.
- **genre:** Genre information for tracks

The project is divided into three categories of problems - **easy, intermediate, and advanced** - each addressing different aspects of the data.

Easy Problems and Solutions:

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

```
SELECT*FROM employee

ORDER BY levels DESC

LIMIT 1
```

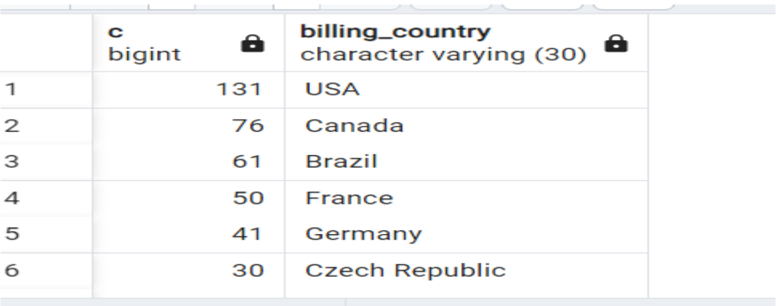


	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (
1	9	Madan	Mohan	Senior General Manager	[null]	L7

Result: This query retrieves the senior-most employee by ordering the employees based on their levels in descending order and selecting the top entry.

Q2)Which countries have the most Invoices?

```
SELECT  COUNT(*) AS c, billing_country

FROM invoice

GROUP BY billing_country

ORDER BY c DESC
```

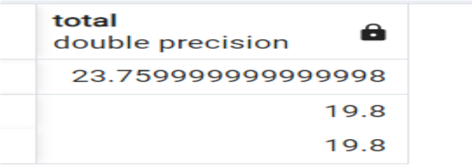


	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic

Result: This query counts the number of invoices per country, groups the results by country, and orders them in descending order of invoice count.

Q3)What are top 3 values of total invoice?

```
SELECT total FROM invoice

ORDER BY total DESC

LIMIT 3
```

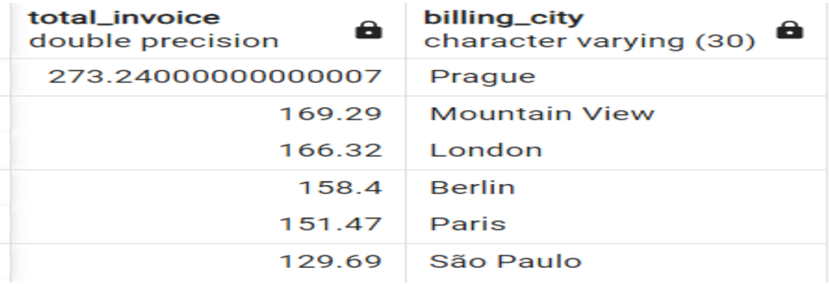


total double precision
23.759999999999998
19.8
19.8

Result: This query retrieves the top 3 highest invoice totals by ordering the invoices in descending order of the total amount.

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.

```
SELECT SUM(total) AS total_invoice ,billing_city

FROM invoice

GROUP BY billing_city

ORDER BY total_invoice DESC
```



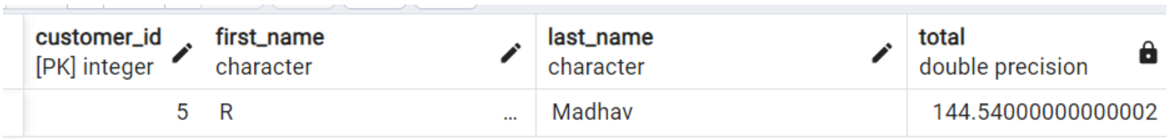
total_invoice double precision	billing_city character varying (30)
273.24000000000007	Prague
169.29	Mountain View
166.32	London
158.4	Berlin
151.47	Paris
129.69	São Paulo

*** Here **Prague** city is the best customer.

Result: This query calculates the sum of all invoice totals for each city, groups the results by city, and orders them by the total invoice amount in descending order.

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.

```
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
5	R	Madhav	144.54000000000002

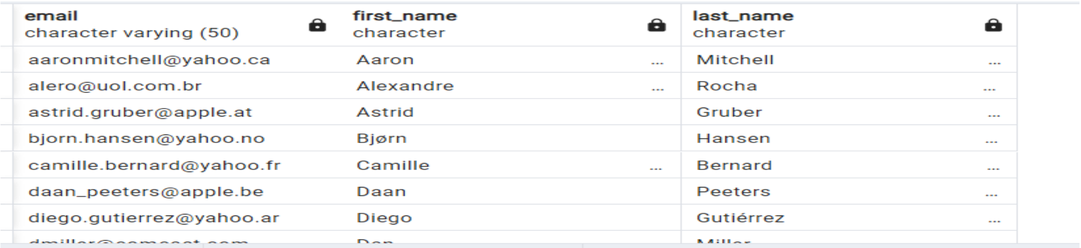
*** R Madhav is the best customer.

Result: This query identifies the customer who has spent the most money by summing the invoice totals for each customer, grouping by customer ID, and ordering by the total spent in descending order.

Intermediate Problems and Solutions:

Q1)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.

```
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 ASC
```

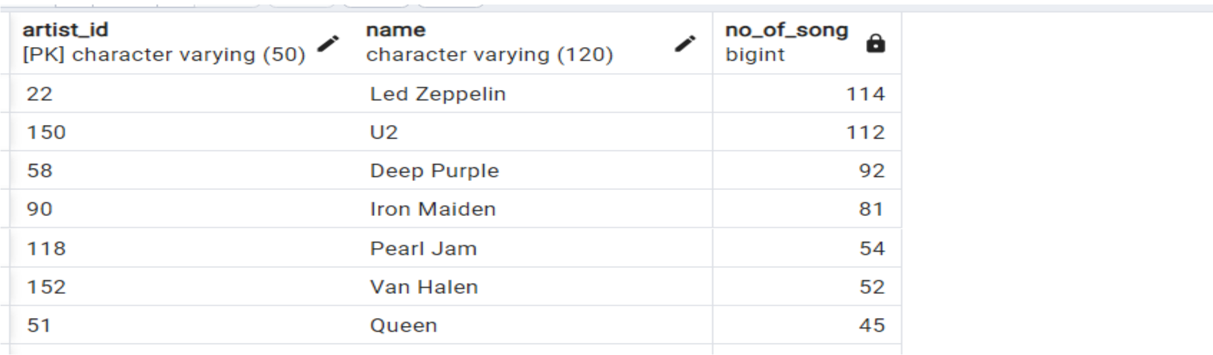


email character varying (50)	first_name character	last_name character
aaronmitchell@yahoo.ca	Aaron	Mitchell
alero@uol.com.br	Alexandre	Rocha
astrid.gruber@apple.at	Astrid	Gruber
bjorn.hansen@yahoo.no	Bjørn	Hansen
camille.bernard@yahoo.fr	Camille	Bernard
daan_peeters@apple.be	Daan	Peeters
diego.gutierrez@yahoo.ar	Diego	Gutiérrez
dmiller@comcast.com	Dan	Miller

Result: This query retrieves distinct customer emails, first names, and last names for those who have purchased Rock music tracks, ordered alphabetically by email.

Q2)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 no_of_song

FROM track

JOIN album ON track.album_id=album.album_id

JOIN artist ON artist.artist_id=album.artist_id

JOIN genre ON genre.genre_id=track.genre_id

WHERE genre.name='Rock'

GROUP BY artist.artist_id

ORDER BY no_of_song  DESC

LIMIT 10
```

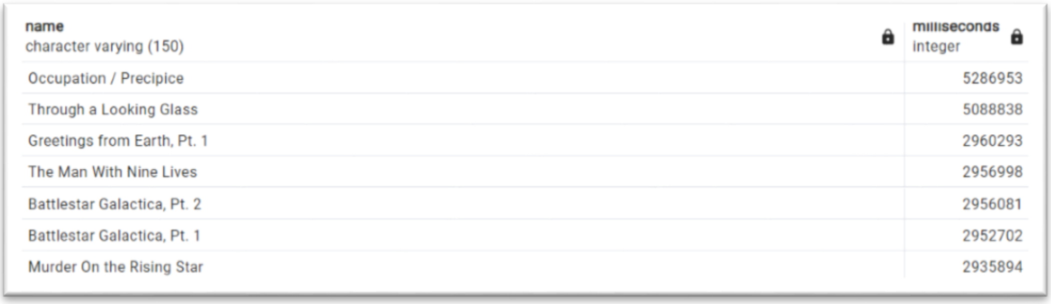


artist_id [PK] character varying (50)	name character varying (120)	no_of_song bigint
22	Led Zeppelin	114
150	U2	112
58	Deep Purple	92
90	Iron Maiden	81
118	Pearl Jam	54
152	Van Halen	52
51	Queen	45

Result: This query identifies the top 10 artists who have the most Rock tracks by counting the number of Rock tracks each artist has, grouping by artist ID, and ordering by the track count in descending order.

Q3) 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) AS avg_track_length
        FROM track
)
ORDER BY milliseconds DESC
```



name character varying (150)	milliseconds integer
Occupation / Precipice	5286953
Through a Looking Glass	5088838
Greetings from Earth, Pt. 1	2960293
The Man With Nine Lives	2956998
Battlestar Galactica, Pt. 2	2956081
Battlestar Galactica, Pt. 1	2952702
Murder On the Rising Star	2935894

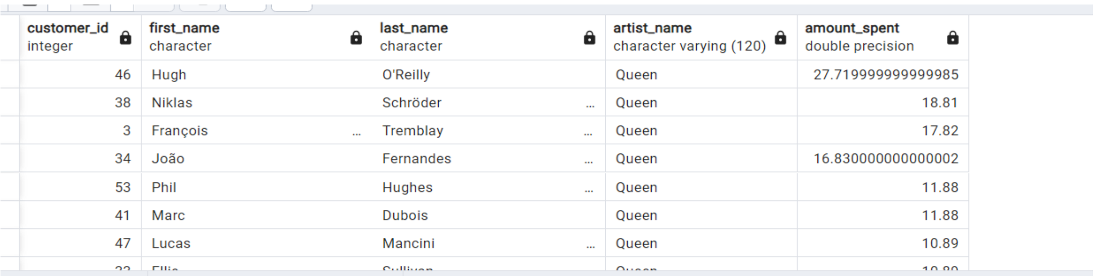
Result: This query retrieves track names and their lengths for tracks longer than the average track length, ordered by length in descending order.

Advanced Problems and Solutions:

Q1)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


        )
```

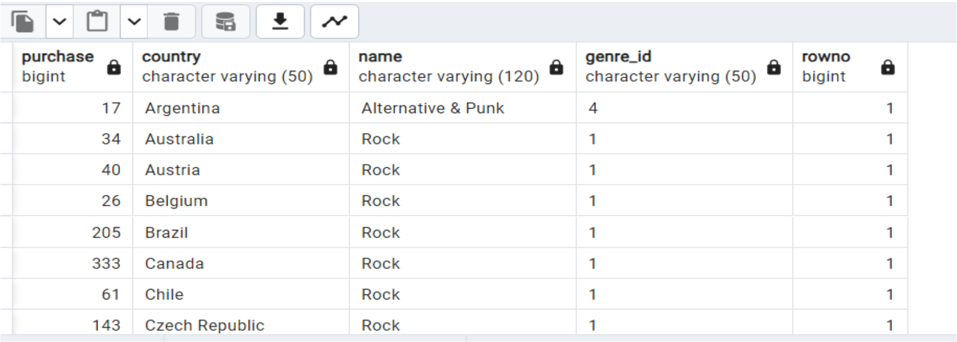


customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
46	Hugh	O'Reilly	Queen	27.719999999999985
38	Niklas	Schröder ...	Queen	18.81
3	François ...	Tremblay ...	Queen	17.82
34	João	Fernandes ...	Queen	16.830000000000002
53	Phil	Hughes ...	Queen	11.88
41	Marc	Dubois	Queen	11.88
47	Lucas	Mancini ...	Queen	10.89
23	Ellie	Sullivan	Queen	10.89

<u>Result:</u> This query calculates the amount spent by each customer on the best-selling artist by summing the unit prices times quantities for each customer-artist pair, grouped by customer and artist.

Q2) 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.

```
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.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
```

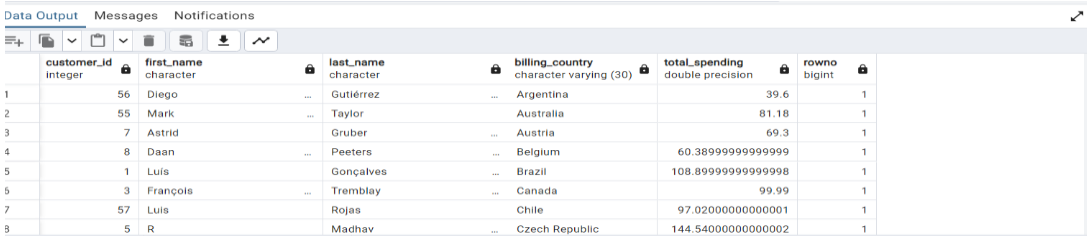


purchase bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
17	Argentina	Alternative & Punk	4	1
34	Australia	Rock	1	1
40	Austria	Rock	1	1
26	Belgium	Rock	1	1
205	Brazil	Rock	1	1
333	Canada	Rock	1	1
61	Chile	Rock	1	1
143	Czech Republic	Rock	1	1

<u>Result:</u> This query determines the most popular genre for each country by counting the number of purchases for each genre within each country, ordering by the count in descending order, and selecting the top genre per country.

Q3) 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.

```
WITH customer_with_country AS(

SELECT customer.customer_id, first_name,last_name,billing_country,SUM(total) AS total_spending,ROW_NUMBER()OVER (PARTITION BY billing_country ORDER BY SUM(total)DESC)AS RowNo

FROM invoice

JOIN customer ON customer.customer_id=invoice.customer_id

GROUP BY 1,2,3,4

ORDER BY 4 ASC,5 DESC)

SELECT* FROM customer_with_country WHERE RowNo<=1
```



	customer_id integer	first_name character	last_name character	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

Result: This query identifies the top-spending customer in each country by summing their total invoice amounts, grouping by customer and country, ordering by the total spent in descending order, and selecting the top spender per country.

Conclusion

The Music Store Data Analysis Project demonstrates the power of SQL in extracting valuable insights from a relational database. By addressing a variety of problems at different difficulty levels, the project provides a comprehensive analysis of the music store's data, helping to identify key employees, top customers, popular music genres, and more. These insights can be used to make informed business decisions and strategize future marketing efforts.