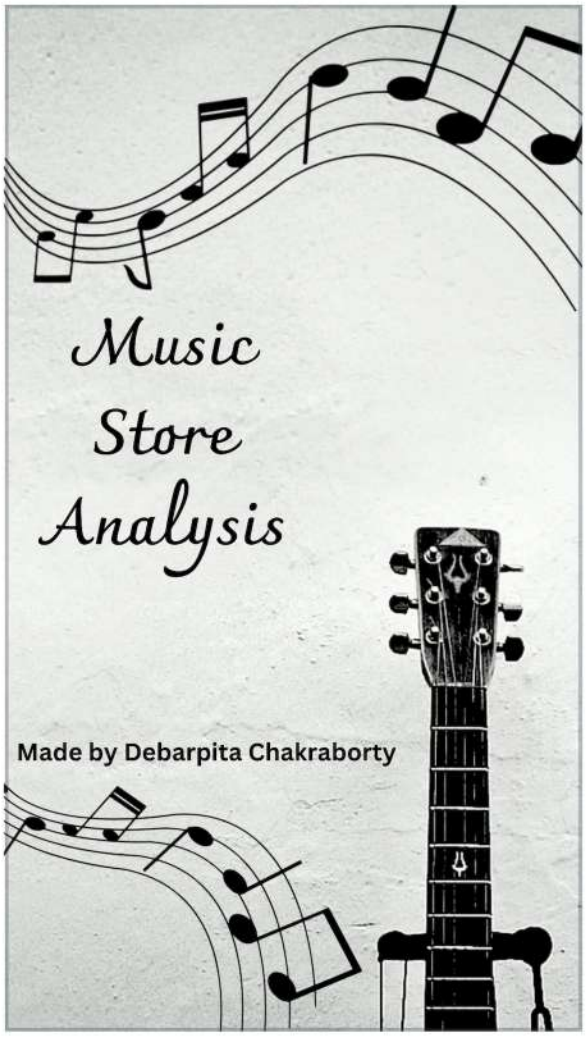
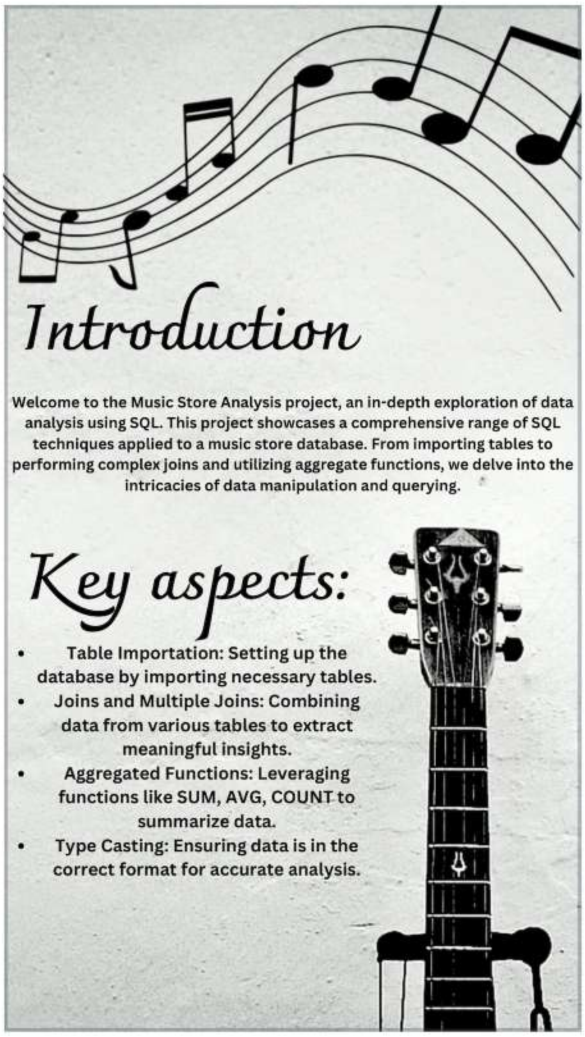


Music Store Analysis

Made by Debarpita Chakraborty

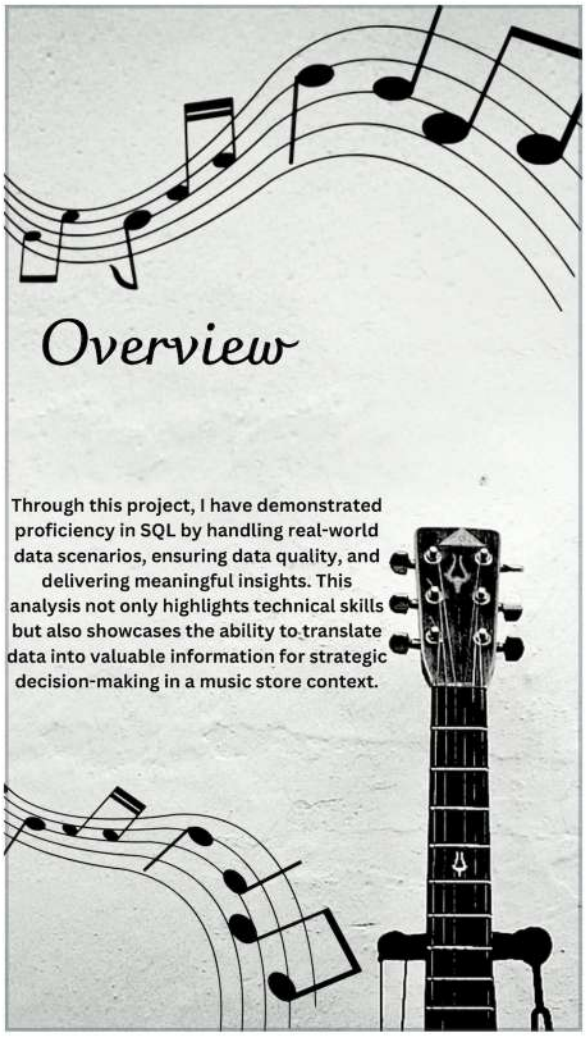


Introduction

Welcome to the Music Store Analysis project, an in-depth exploration of data analysis using SQL. This project showcases a comprehensive range of SQL techniques applied to a music store database. From importing tables to performing complex joins and utilizing aggregate functions, we delve into the intricacies of data manipulation and querying.

Key aspects:

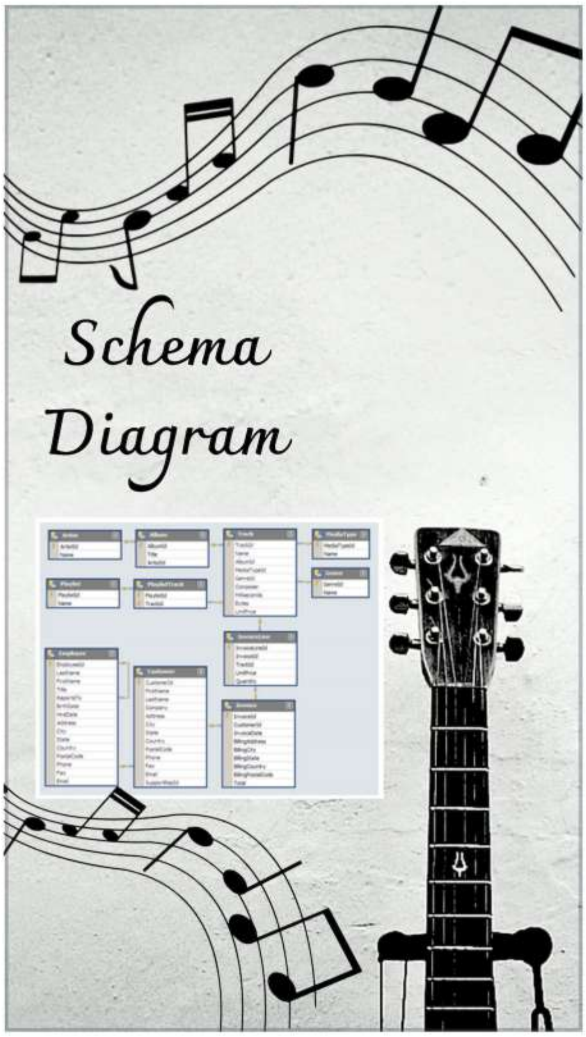
- Table Importation: Setting up the database by importing necessary tables.
- Joins and Multiple Joins: Combining data from various tables to extract meaningful insights.
- Aggregated Functions: Leveraging functions like SUM, AVG, COUNT to summarize data.
- Type Casting: Ensuring data is in the correct format for accurate analysis.

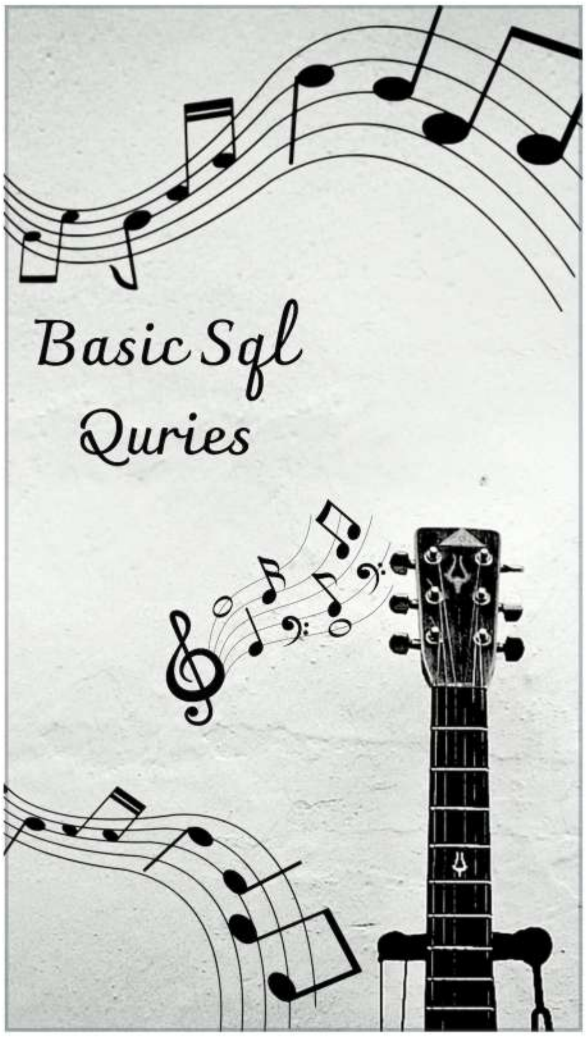


Overview

Through this project, I have demonstrated proficiency in SQL by handling real-world data scenarios, ensuring data quality, and delivering meaningful insights. This analysis not only highlights technical skills but also showcases the ability to translate data into valuable information for strategic decision-making in a music store context.

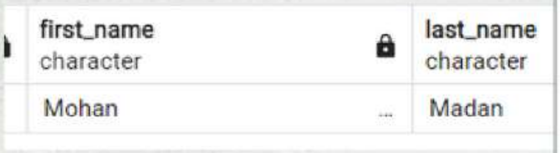
Schema Diagram



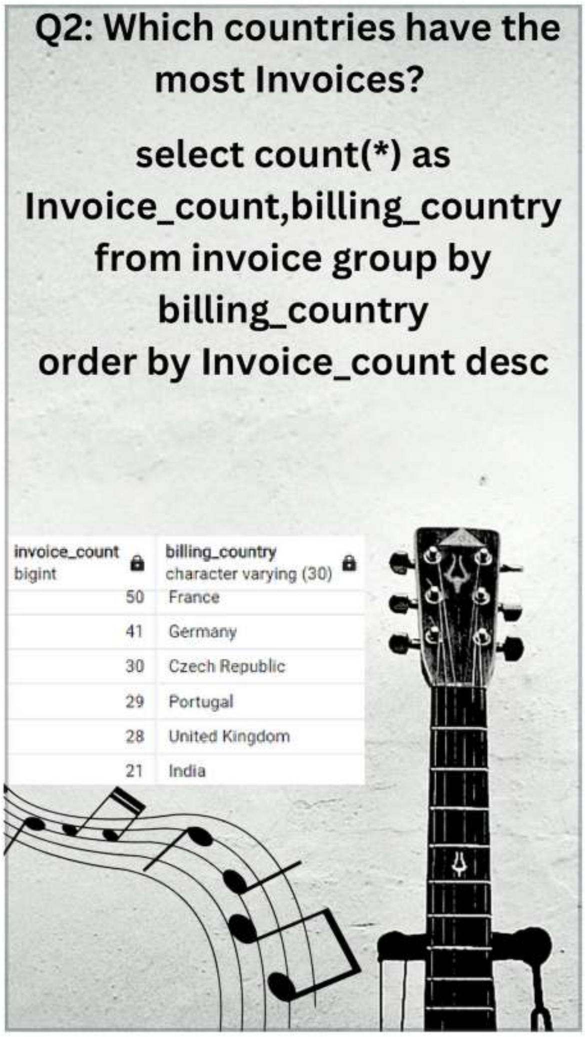

Basic Sql
Quries

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

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



first_name character	last_name character
Mohan	Madan



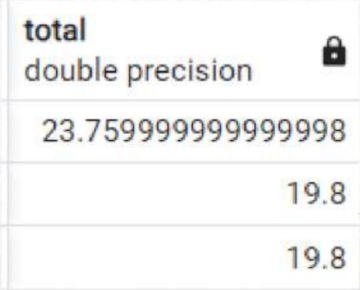
Q2: Which countries have the most Invoices?

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

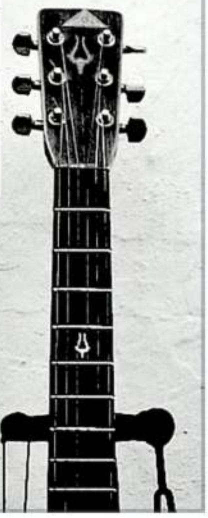
invoice_count bigint	billing_country character varying (30)
50	France
41	Germany
30	Czech Republic
29	Portugal
28	United Kingdom
21	India

Q3: What are top 3 values of total invoice?

```
select total from invoice
order by total desc
limit 3
```

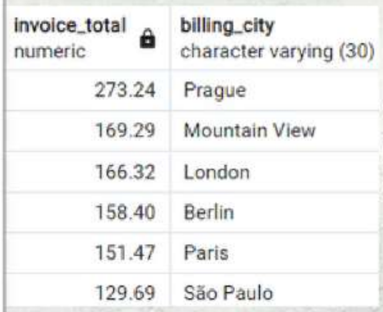


total double precision
23.759999999999998
19.8
19.8



Q4: Which city has the best customers?

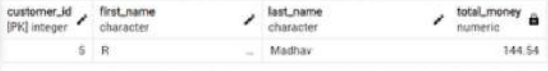
```
select sum(total) as
Invoice_Total,
billing_city from invoice group
by billing_city
order by Invoice_Total desc
```



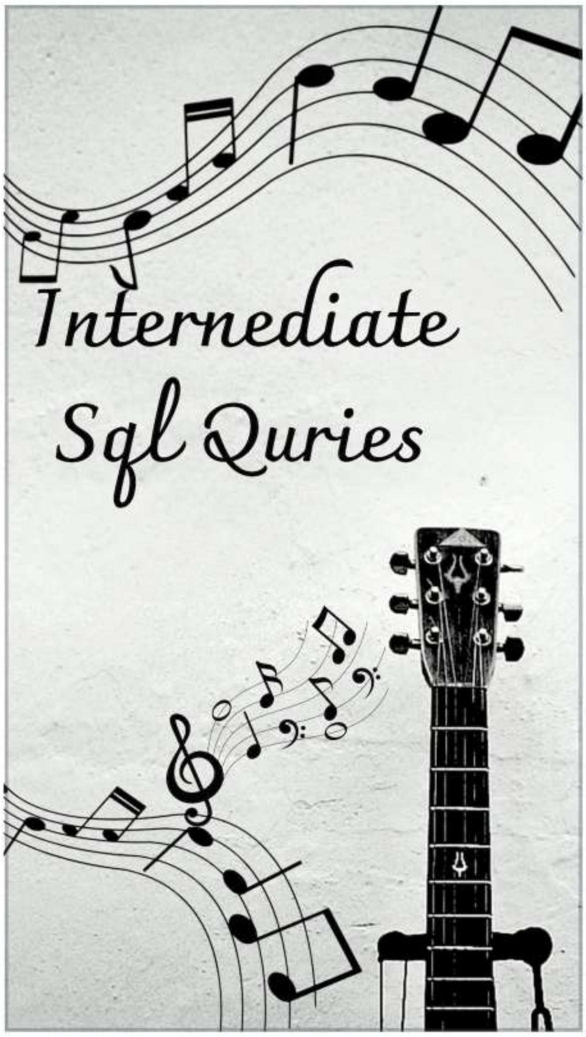
invoice_total numeric 🔒	billing_city character varying (30)
273.24	Prague
169.29	Mountain View
166.32	London
158.40	Berlin
151.47	Paris
129.69	São Paulo

Q5: Who is the best customer?

```
select
customer.customer_id,customer.first_name,customer.last_name,
round(CAST(sum(invoice.total) AS numeric), 2)
as Total_Money
from customer join
invoice on customer.customer_id =
invoice.customer_id
group by customer.customer_id
order by Total_Money desc
limit 1
```

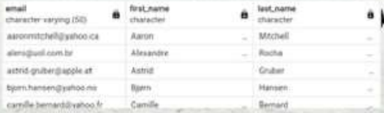


customer_id [PK] integer	first_name character	last_name character	total_money numeric
5	R	Madhav	144.54


Internediate
Sql Quries

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

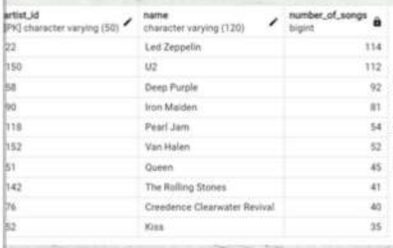
```
SELECT DISTINCT email,first_name, last_name
FROM customer
JOIN invoice ON customer.customer_id =
invoice.customer_id
JOIN invoice_line ON invoice.invoice_id =
invoice_line.invoice_id
WHERE track_id IN(
SELECT track_id FROM track
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock'
)
ORDER BY email
```



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

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.

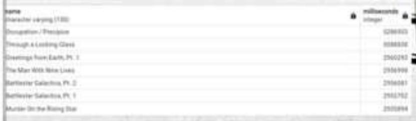
```
SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS
number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 10
```



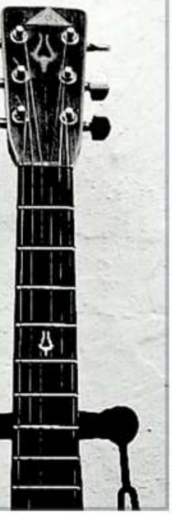
artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
22	Led Zeppelin	114
150	U2	112
58	Deep Purple	92
90	Iron Maiden	81
118	Pearl Jam	54
152	Van Halen	52
51	Queen	45
142	The Rolling Stones	41
76	Creedence Clearwater Revival	40
52	Kiss	35

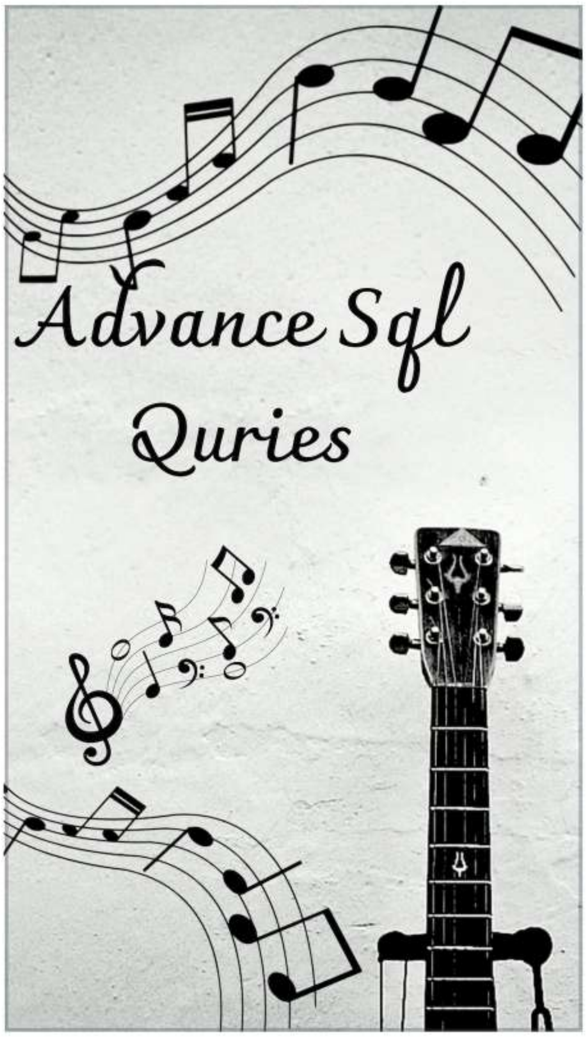
Q8: Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

```
SELECT name,track.milliseconds
FROM track
WHERE milliseconds > (
SELECT AVG(milliseconds) AS
avg_track_length
FROM track )
ORDER BY milliseconds DESC
```



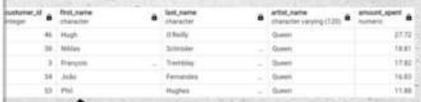
name character varying (150)	milliseconds integer
Occupation / Precipice	5286953
Through a Looking Glass	5088838
Greetings from Earth, Pt. 1	2960293
The Man With Nine Lives	2956998
Battlestar Galactica, Pt. 2	2956081
Battlestar Galactica, Pt. 1	2952702
Murder On the Rising Star	2935894



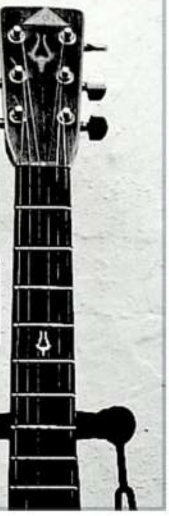

Advance Sql
Quries

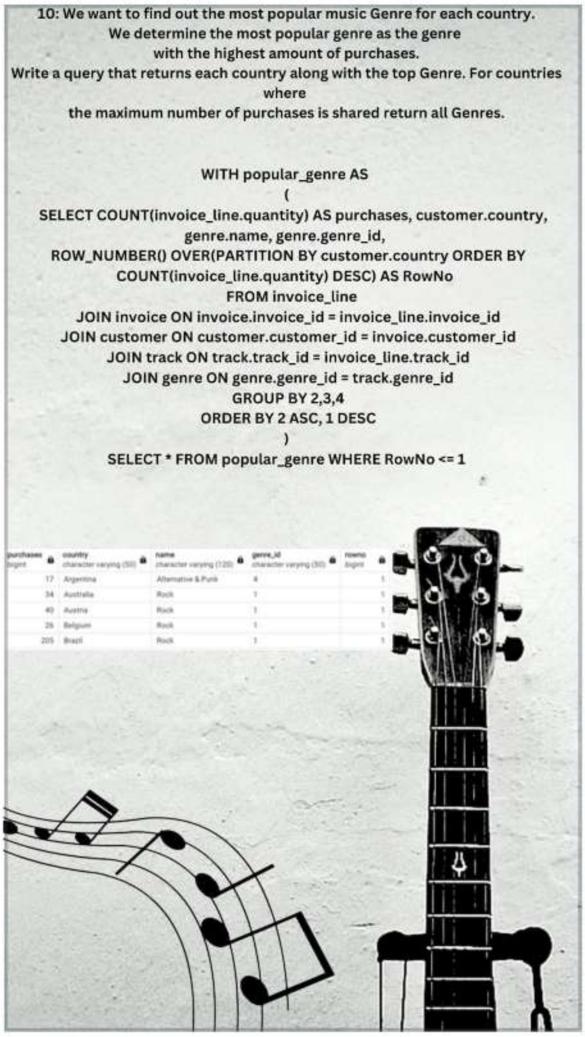
Q9: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

```
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,
round(CAST(SUM(il.unit_price*il.quantity) As
numeric),2) 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 numeric
46	Hugh	O'Reilly	Queen	27.72
38	Niklas	Schröder	Queen	18.81
3	François	Tremblay	Queen	17.82
34	João	Fernandes	Queen	16.83
53	Phil	Hughes	Queen	11.88

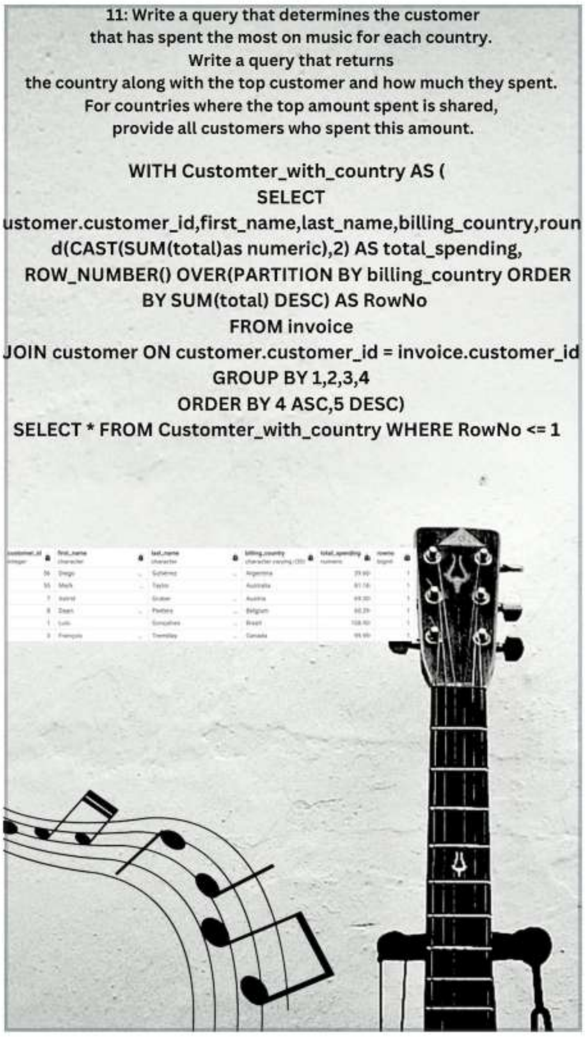




10: We want to find out the most popular music Genre for each country.
We determine the most popular genre as the genre
with the highest amount of purchases.
Write a query that returns each country along with the top Genre. For countries
where
the maximum number of purchases is shared return all Genres.

```
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 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
```

purchases 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



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.

```
WITH Customter_with_country AS (
SELECT
ustomer.customer_id,first_name,last_name,billing_country,roun
d(CAST(SUM(total)as numeric),2) 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 Customter_with_country WHERE RowNo <= 1
```

customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending numeric	rowno bigint
56	Diego	Gutiérrez	Argentina	39.60	1
55	Mark	Taylor	Australia	81.18	1
7	Astrid	Gruber	Austria	69.30	1
8	Daan	Peeters	Belgium	60.39	1
1	Luís	Gonçalves	Brazil	108.90	1
3	François	Tremblay	Canada	99.99	1

Thank
You