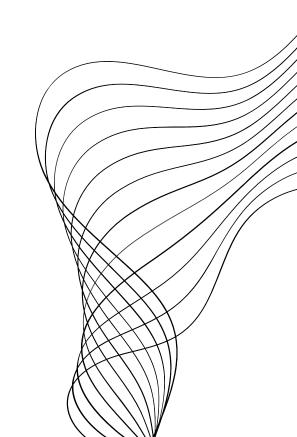


MUSIC DATABASE ANALYSIS

SQL PROJECT





ABOUT ME

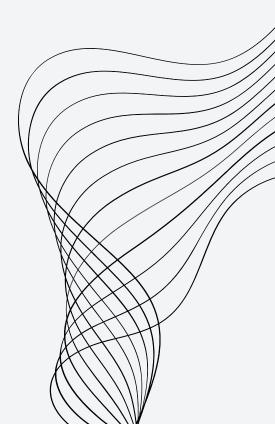
My name is Sahil Gawai a fresher data analyst with a strong foundation in SQL and data analysis. I have worked on a project where I utilized SQL queries to solve questions related to a music playlist analysis on a music playlist database. This project helped me develop my skills in data extraction, transformation, and analysis, enabling me to uncover valuable insights from complex datasets.



Q1: WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

```
3
4 • SELECT title, last_name, first_name
5 FROM employee
6 ORDER BY levels DESC
7 LIMIT 1;
8
```

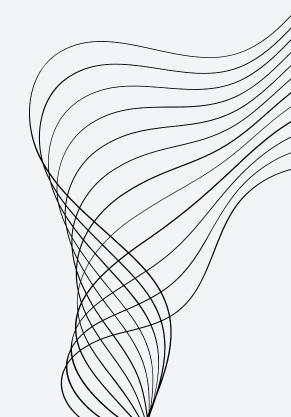
Result Grid				
	title	last_name	first_name	
•	General Manager	Adams	Andrew	
emplovee 1 ×				
A .				



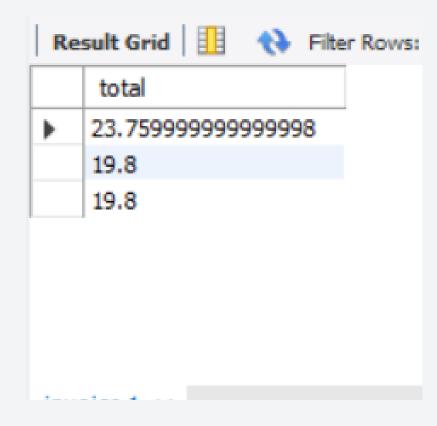
Q2: WHICH COUNTRIES HAVE THE MOST INVOICES?

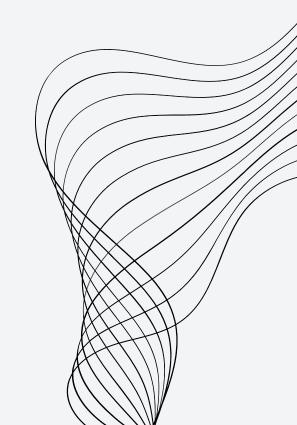
```
12 • SELECT COUNT(*) AS c, billing_country
13 FROM invoice
14 GROUP BY billing_country
15 ORDER BY c DESC;
16
```

Result Grid				
	С	billing_country		
•	131	USA		
	76	Canada		
	61	Brazil		
	50	France		
	41	Germany		
Res	ult3 ×			









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

```
29 • SELECT billing_city,SUM(total) AS InvoiceTotal

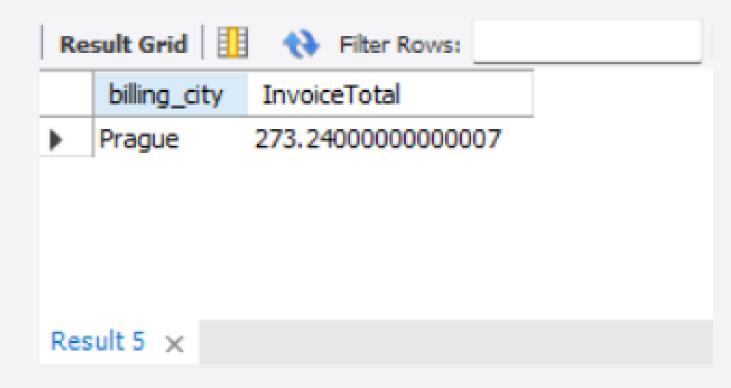
30 FROM invoice

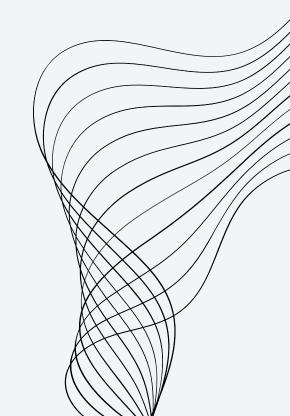
31 GROUP BY billing_city

32 ORDER BY InvoiceTotal DESC

33 LIMIT 1;

34
```





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

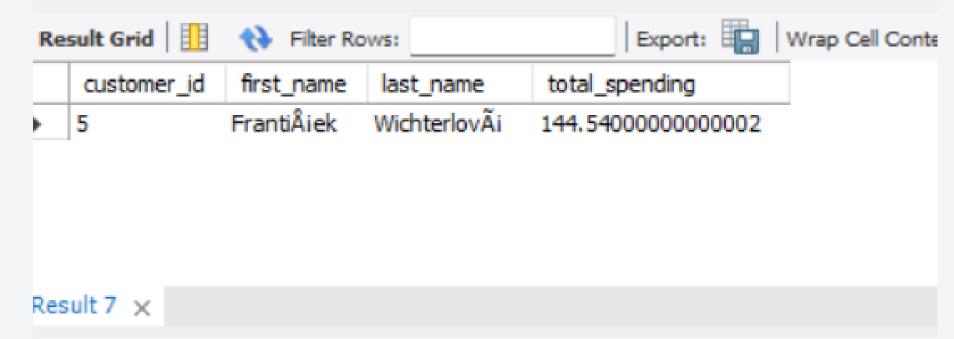
```
SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending
FROM customer

JOIN invoice ON customer.customer_id = invoice.customer_id

GROUP BY customer.customer_id, first_name, last_name

ORDER BY total_spending DESC

LIMIT 1;
```

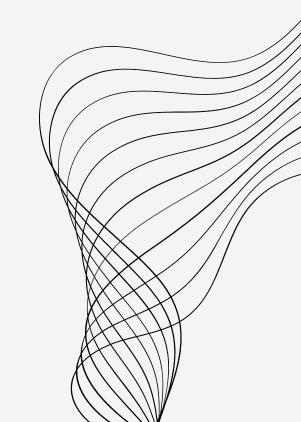


Q6: WRITE QUERY TO RETURN THE EMAIL, FIRST NAME, LAST NAME, & GENRE OF ALL ROCK MUSIC LISTENERS. RETURN YOUR LIST ORDERED ALPHABETICALLY BY EMAIL STARTING WITH A.

```
SELECT DISTINCT email, first_name, last_name
51
      FROM customer
      JOIN invoice ON customer.customer_id = invoice.customer_id
52
      JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
53

    WHERE track id IN(
54
          SELECT track_id FROM track
55
          JOIN genre ON track.genre_id = genre.genre_id
56
          WHERE genre.name LIKE 'Rock'
57
58
      ORDER BY email;
```

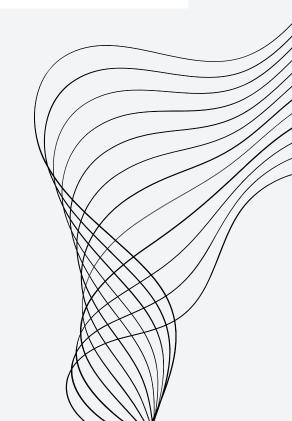
Re	sult Grid 🔢 🔷 Filter Rows:		Export:
	email	first_name	last_name
>	aaronmitchell@yahoo.ca	Aaron	Mitchell
	alero@uol.com.br	Alexandre	Rocha
	astrid.gruber@apple.at	Astrid	Gruber
	bjorn.hansen@yahoo.no	Bjà ˌrn	Hansen
	camille.bernard@yahoo.fr	Camille	Bernard
	daan_peeters@apple.be	Daan	Peeters
	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
	dmiller@comcast.com	Dan	Miller
Res	dominiquelefehvre@amail.com sult 9 ×	Dominia: ie	l efehvre



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.

```
63
64 • SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
65 FROM track
66 JOIN album2 ON album2.album_id = track.album_id
67 JOIN artist ON artist.artist_id = album2.artist_id
68 JOIN genre ON genre.genre_id = track.genre_id
69 WHERE genre.name LIKE 'Rock'
70 GROUP BY artist.artist_id,artist.name
71 ORDER BY number_of_songs DESC
72 LIMIT 10;
```

	artist_id	name	number_of_songs
•	1	AC/DC	18 18
	3	Aerosmith	15
	8	Audioslave	14
	22	Led Zeppelin	14
	4	Alanis Morissette	13
	5	Alice In Chains	12
	23	Frank Zappa & Captain Beefheart	9
	2	Accept	4



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.

```
77 • SELECT * FROM track;

78 • SELECT name, milliseconds

79 FROM track

80 • WHERE milliseconds > (

81 SELECT AVG(milliseconds) AS avg_track_length

82 FROM track )

83 ORDER BY milliseconds DESC;
```

Result Grid	Filter Rows:		Export:
name		milliseconds	
No More Tears		555075	
The Alchemist	Wheels Of Conf	509413	
Wheels Of Confusio	n / The Straighten	er 494524	
Book Of Thel		494393	
You Oughta Know (Alternate)	491885	
Terra		482429	
Snoopy's search-Re	d baron	456071	
Sozinho (Hitmakers	Sozinho (Hitmakers Classic Mix)		
Master Of Puppets		436453	
Stone Crazy		433397	
track 15 ×			

