

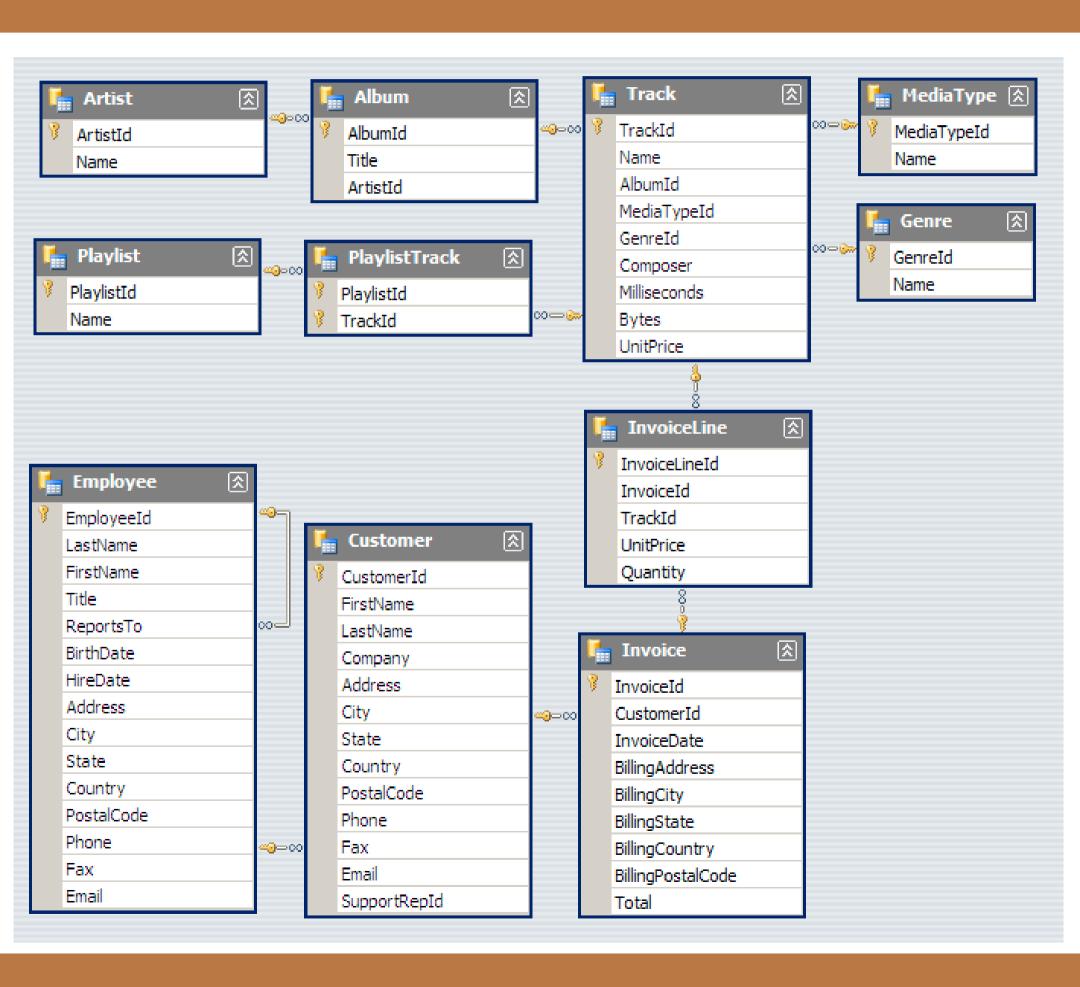
MUSIC STORE DATA ANALYSIS Using SQL



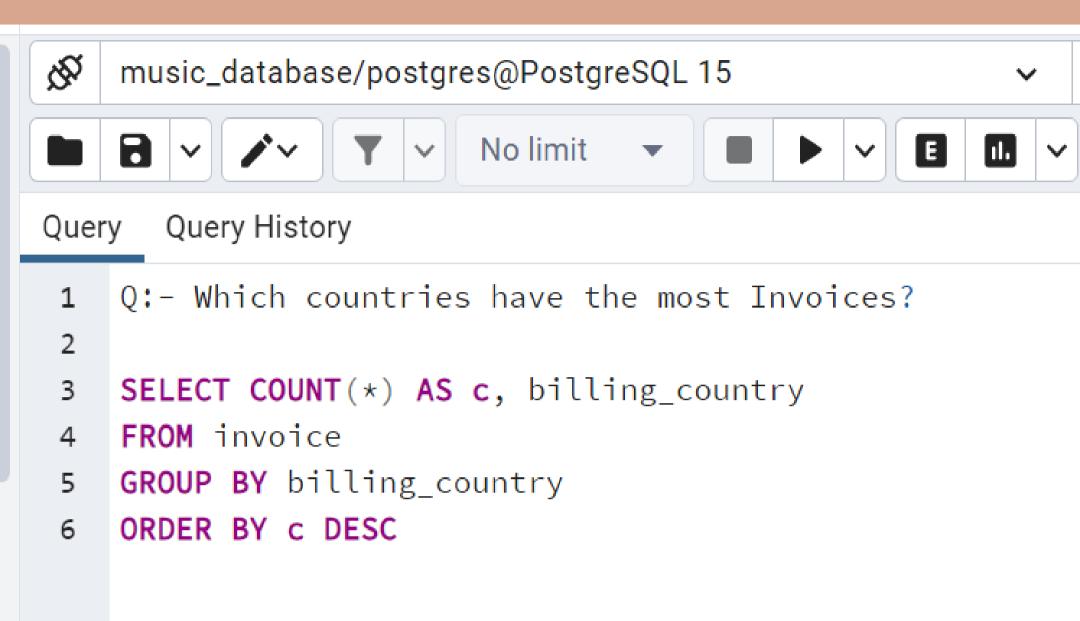
OBJECTIVE:Project FOR FINDING OUT SOMEUSEFUL INSIGHTS USING SQL FOR BUSINESS GROWTH OF THE MUSIC STORE

Understanding the Dataset:

The Music Store dataset contains total 11tables. The schema diagram for this dataset is shown below.

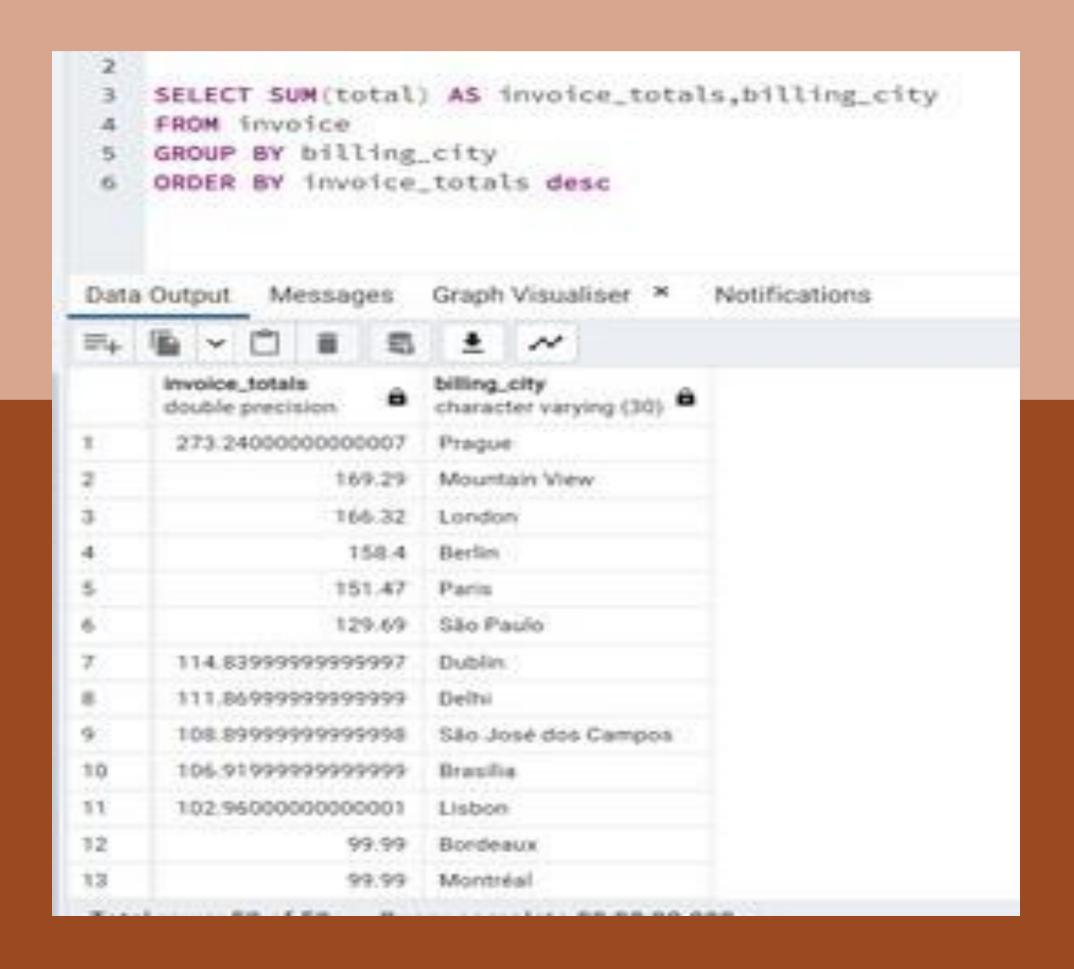


WHICH COUNTRIES HAVE THE MOST INVOICES?



Data	Data Output Messages Notifications							
=+		~ = =						
	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						
フ	29	Portugal						
8	28	United Kingdom						
9	21	India						
10	13	Chile						

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.



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
    SELECT DISTINCT email, first_name, last_name
 1
    FROM customer
 2
    JOIN invoice ON customer.customer_id = invoice.customer_id
 3
    JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
 4
    WHERE track id IN(
 5
 6
        SELECT track_id FROM track
        JOIN genre ON track.genre_id = genre.genre_id
 7
        WHERE genre.name LIKE 'Rock'
 8
 9
    ORDER BY email;
10
```

	email character varying (50)	first_name character (50)	â	last_name character (50)	â	genre character varying (
1	aaronmitchell@yahoo.ca	Aaron	***	Mitchell	***	Rock
2	alero@uol.com.br	Alexandre	***	Rocha		Rock
3	astrid.gruber@apple.at	Astrid	1	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
9	dominiquelefebvre@gmail.c	Dominique	***	Lefebvre	***	Rock

LET'S INVITE THE ARTISTS WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR DATASET. WRITE A QUERY THAT RETURNS THE ARTISTNAME AND TOTAL TRACK COUNT OF THE TOP 10 ROCK BANDS

```
Query
      Query History
    SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
1
    FROM track
2
    JOIN album ON album.album_id = track.album_id
3
    JOIN artist ON artist.artist_id = album.artist_id
4
    JOIN genre ON genre genre id = track genre id
5
    WHERE genre.name LIKE 'Rock'
6
    GROUP BY artist.artist_id
7
    ORDER BY number_of_songs DESC
8
    LIMIT 10;
9
```

=+ □		<u>*</u>	
	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

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

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
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
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008

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;
                           Notifications
Data Output
              Messages
=+
      customer_id
                     first_name
                                                 last_name
                                                                              artist_name
                                                                                                     amount_spent
                                                                              character varying (120)
      integer
                                                 character
                                                                                                     double precision
                     character
                                                  O'Reilly
                                                                                                      27.71999999999985
1
                 46
                     Hugh
                                                                              Queen
2
                 38
                     Niklas
                                                  Schröder
                                                                              Queen
                                                                                                                    18.81
3
                 3
                     François
                                                 Tremblay
                                                                              Queen
                                                                                                                    17.82
                                                  Fernandes
                                                                                                      16.830000000000002
4
                 34
                     João
                                                                              Queen
5
                 53
                     Phil
                                                 Hughes
                                                                              Queen
                                                                                                                    11.88
                 41
                                                 Dubois
6
                     Marc
                                                                              Oueen
                                                                                                                    11.88
7
                 47
                                                  Mancini
                                                                                                                    10.89
                     Lucas
                                                                              Queen
```

Sullivan

Madhav

Miller

Queen

Queen

Queen

10.89

3.96

3.96

Ellie

Dan

R

33

20

5

8

10

WE WANT TO FIND OUT THE MOST POPULAR MUSIC GENRE FOR EACH COUNTRY. WE DETERMINE THE MOST POPULAR GENRE AS THE GENREWITH THE HIGHEST AMOUNT OF PURCHASES. WRITE A QUERY THAT RETURNS EACH COUNTRY A LONG WITH THE TOP GENRE. FOR COUNTRIES WHERETHE 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</pre>
```

SELECT	SELECT * FROM popular_genre WHERE RowNo <= 1										
Data Output Messages Notifications											
=+ 「	~ ° ~										
	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						
7	61	Chile	Rock	1	1						
8	143	Czech Republic	Rock	1	1						
9	24	Denmark	Rock	1	1						
10	46	Finland	Rock	1	1						

WRITE A QUERY THAT DETERMINES THE CUSTOMER THAT HAS SPENT THE MOST ONFOR EACH COUNTRY.WRITE A QUERY THAT RETURNS THE COUNTRY MUSIC 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 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 Customter_with_country WHERE RowNo <= 1</pre>
```

Data Output	Messages	Notifications		

Data	Data Output Messages Notifications									
-i+										
	customer_id integer	first_name character	â	last_name character	â	billing_country character varying (30)	total_spending double precision	rowno bigint	â	
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	108.8999999999998		1	
6	3	François .		Tremblay		Canada	99.99		1	
7	57	Luis		Rojas		Chile	97.0200000000001		1	
8	5	R		Madhav		Czech Republic	144.540000000000002		1	
9	9	Kara		Nielsen		Denmark	37.61999999999999		1	
10	44	Terhi		Hämäläinen		Finland	79.2		1	

THANK YOU