

The background features a light purple-to-blue gradient. Scattered across the surface are numerous realistic water droplets of varying sizes, some with highlights and shadows. A faint, large, light-colored circular graphic is centered behind the text.

SQL PROJECT

on

Music_store_analysis

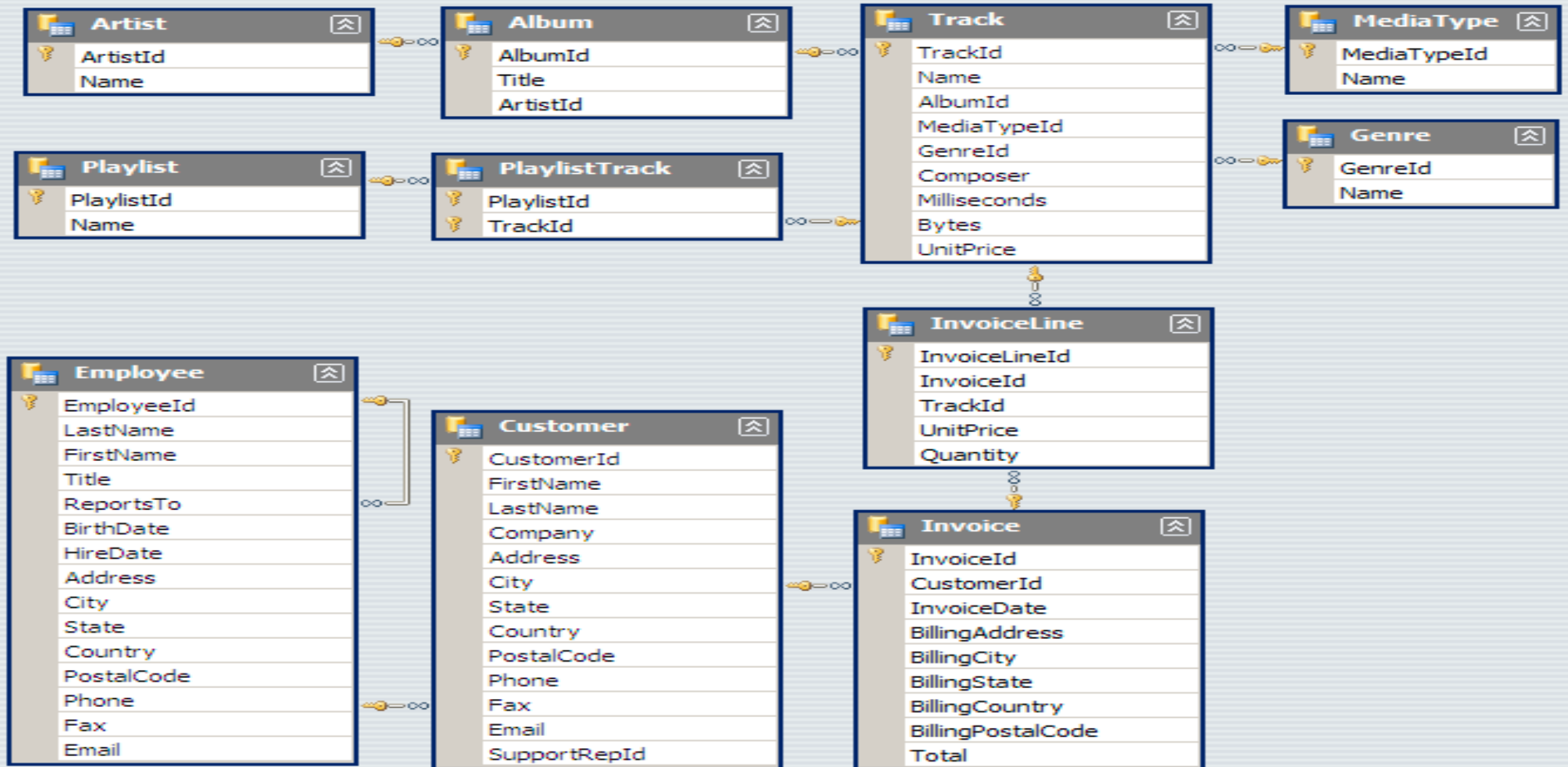
Database used PostgreSQL

Tool used pgAdmin4

About Project

In this guided project, we will be working with a database called `music_store_database` which contains information about a fictional music store. The database includes several tables on invoice information, track, album, artist, and genre data, and employee and customer information related to the store's sales. We will use this PostgreSQL database and pgAdmin4 tool in order to explore and analyze three fictional sets of Business questions and Propositions. Questions consist of three sets(easy, moderate, and advanced). Each set of levels consists of questions with their answers and images of the following output.

No. of Tables



Question Set 1 - Easy

1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are the top 3 values of the total invoice?
4. 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.
5. 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.

Answers Set 1 – Easy

Ans1. Query with the following output image

The screenshot shows the pgAdmin 4 interface with the following components:

- Object Explorer:** Lists database objects including FTS Configurations, Dictionaries, Parsers, Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (11), Subscriptions, and the postgres database.
- Query Editor:** Contains the following SQL query:

```
1 /* Question Set 1 - Easy */
2
3 /* Q1: Who is the senior most employee based on job title? */
4
5 Select * From employee
6 ORDER BY levels DESC
7 LIMIT 1;
```
- Data Output:** Displays the result of the query in a table with 8 columns: employee_id, last_name, first_name, title, reports_to, levels, and birthdate. The output shows one row for the Senior General Manager.
- Bottom Panel:** Shows the status "Total rows: 1 of 1" and "Query complete 00:00:00.074".

employee_id	last_name	first_name	title	reports_to	levels	birthdate
9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00

Ln 7, Col 9

Ans2. Query with the following output image

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > public. The main panel displays a SQL query in the Query editor:

```
/* Q2: Which countries have the most Invoices? */  
SELECT COUNT(*) AS c, billing_country  
FROM invoice  
GROUP BY billing_country  
ORDER BY c DESC
```

The Data Output tab shows the results of the query:

	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
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia

Total rows: 24 of 24 Query complete 00:00:00.143 Ln 16, Col 16

Ans3. Query with the following output image

The screenshot displays the pgAdmin 4 web interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > Schemas (1) > public. The main panel shows the 'Query' tab with the following SQL query:

```
1
2
3  /* Q3: What are top 3 values of total invoice? */
4
5  SELECT total FROM invoice
6  ORDER BY total DESC
7  LIMIT 3;
```

Below the query editor, the 'Data Output' tab shows the results of the query in a table format:

	total	
	double precision	
1	23.759999999999998	
2		19.8
3		19.8

The status bar at the bottom indicates 'Total rows: 3 of 3' and 'Query complete 00:00:00.071'. The bottom of the image shows a Windows taskbar with various icons and system information: 36°C Partly sunny, 19:45, 19-06-2023.

Ans4. Query with the following output image

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > public. The main query editor shows a SQL query to find the city with the highest sum of invoice totals. The query is as follows:

```
/* Q4: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made
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 invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC;
```

The query results are displayed in the Data Output tab, showing a table with two columns: invoice_total (double precision) and billing_city (character varying (30)). The results are sorted by invoice_total in descending order.

	invoice_total double precision	billing_city character varying (30)
1	273.240000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.839999999999997	Dublin
8	111.869999999999999	Delhi
9	108.899999999999998	São José dos Campos
10	106.919999999999999	Brasília
11	102.960000000000001	Lisbon
12	99.99	Bordeaux

Total rows: 53 of 53 Query complete 00:00:00.075 Ln 30, Col 29

Ans5. Query with the following output image

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > public. The main pane shows the SQL editor with the following query:

```
31
32 /* Q5: Who is the best customer? The customer who has spent the most money will be declared the best customer.
33 Write a query that returns the person who has spent the most money.*/
34
35 SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending
36 FROM customer
37 JOIN invoice ON customer.customer_id = invoice.customer_id
38 GROUP BY customer.customer_id
39 ORDER BY total_spending DESC
40 LIMIT 1;
```

The Data Output tab shows the result of the query:

	customer_id [PK] integer	first_name character	last_name character	total_spending double precision
1	5	R	Madhav	144.540000000000002

At the bottom, the status bar indicates "Total rows: 1 of 1" and "Query complete 00:00:00.078". The bottom of the image shows a Windows taskbar with the search bar, taskbar icons, and system tray information including temperature (35°C), weather (Partly cloudy), and time (19:59, 19-06-2023).

Question Set 2 – Moderate

1. Write a 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.
2. 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.
3. Return all the track names with a song length longer than the average. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

Answers Set 2 - Moderate

Ans1. Query with the following output Image

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with the 'public' schema selected. The main pane shows a SQL query for 'music_store_database/postgres@PostgreSQL 15'. The query is as follows:

```
/* Question Set 2 - Moderate */
/* 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. */
/*Method 1 */
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;
```

The 'Data Output' pane shows the results of the query, displaying 6 rows of data. The status bar indicates 'Total rows: 59 of 59' and 'Query complete 00:00:00.074'.

	email character varying (50)	firstname character	lastname character	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

Ln 63, Col 1

Ans2. Query with the following output image

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > public > Tables (11). The main panel displays a SQL query in the Query editor, and the Data Output panel shows the results of the query.

Query:

```
/* 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 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;
```

Data Output:

	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	Credence Clearwater Revival	40

Total rows: 10 of 10 Query complete 00:00:00.087 Ln 86, Col 1

Ans3. Query with the following output image

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 15 > Databases (2) > music_store_database > public. The main pane shows a SQL query for the music_store_database/postgres@PostgreSQL 15 connection. The query is as follows:

```
/* 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;
```

The Data Output pane shows the results of the query, displaying 10 rows of track names and their durations in milliseconds, ordered from longest to shortest. The status bar indicates 'Total rows: 494 of 494' and 'Query complete 00:00:00.103'.

	name	milliseconds
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

Question Set 3 – Advance

1. Find how much amount spent by each customer on artists? Write a query to return the customer name, artist name, and total spent.
2. 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.
3. 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.

Answers Set 3 - Advance

Ans1. Query with the following output Image

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'music_store_database/postgres@PostgreSQL 15*' connection. The query is as follows:

```
100 /* Question Set 3 - Advance */
101 /* Q1: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent. */
102
103 WITH best_selling_artist AS (
104     SELECT artist.artist_id AS artist_id, artist.name AS artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
105     FROM invoice_line
106     JOIN track ON track.track_id = invoice_line.track_id
107     JOIN album ON album.album_id = track.album_id
108     JOIN artist ON artist.artist_id = album.artist_id
109     GROUP BY 1
110     ORDER BY 3 DESC
111     LIMIT 1
112 )
113 SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
114 FROM invoice i
115 JOIN customer c ON c.customer_id = i.customer_id
116 JOIN invoice_line il ON il.invoice_id = i.invoice_id
117 JOIN track t ON t.track_id = il.track_id
118 JOIN album alb ON alb.album_id = t.album_id
119 JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
120 GROUP BY 1,2,3,4
121 ORDER BY 5 DESC;
122
```

The results are displayed in the 'Data Output' tab, showing a table with 6 columns: customer_id, first_name, last_name, artist_name, and amount_spent. The table contains 3 rows of data.

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	Francois	Tremblay	Queen	17.82

Total rows: 43 of 43 Query complete 00:00:00.061

Ln 104, Col 68

Ans2. Query with the following output image

The screenshot shows the pgAdmin 4 interface with the 'music_store_database/postgres@PostgreSQL 15*' connection selected. The 'Query' tab is active, displaying a SQL query that finds the most popular music genre for each country based on the number of purchases. The query uses a CTE named 'popular_genre' to rank genres by purchase count within each country and then selects the top genre for each country.

```
129 /* Q2: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre
130 with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where
131 the maximum number of purchases is shared return all Genres. */
132 /* Steps to Solve: There are two parts in question- first most popular music genre and second need data at country level
133
134 WITH popular_genre AS
135 (
136     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
137     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
138 FROM invoice_line
139 JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
140 JOIN customer ON customer.customer_id = invoice.customer_id
141 JOIN track ON track.track_id = invoice_line.track_id
142 JOIN genre ON genre.genre_id = track.genre_id
143 GROUP BY 2,3,4
144 ORDER BY 2 ASC, 1 DESC
145 )
146 SELECT * FROM popular_genre WHERE RowNo <= 1
147
```

The 'Data Output' tab shows the results of the query, displaying a table with 6 columns: purchases, country, name, genre_id, and rowno. The results show the top genre for each country, with Brazil having the highest number of purchases (205) for the 'Rock' genre.

	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

Total rows: 24 of 24 Query complete 00:00:00.056 Ln 145, Col 2

Ans3. Query with the following output image

The screenshot shows the pgAdmin 4 interface with the 'music_store_database/postgres@PostgreSQL 15*' connection selected. The 'Query' tab is active, displaying a SQL query that determines the top customer for each country based on total spending. The 'Data Output' tab shows the results of the query, which includes columns for customer_id, first_name, last_name, billing_country, total_spending, and rowno. The results are sorted by total_spending in descending order.

```
147
148 /* Q3: Write a query that determines the customer that has spent the most on music for each country.
149 Write a query that returns the country along with the top customer and how much they spent.
150 For countries where the top amount spent is shared, provide all customers who spent this amount. */
151 /* Steps to Solve: Similar to the above question. There are two parts in question-
152 first find the most spent on music for each country and second filter the data for respective customers. */
153
154 WITH Customer_with_country AS (
155     SELECT customer.customer_id, first_name, last_name, billing_country, SUM(total) AS total_spending,
156            ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
157     FROM invoice
158     JOIN customer ON customer.customer_id = invoice.customer_id
159     GROUP BY 1,2,3,4
160     ORDER BY 4 ASC,5 DESC)
161 SELECT * FROM Customer_with_country WHERE RowNo <= 1
162
```

	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	Luis	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

Total rows: 24 of 24 Query complete 00:00:00.096 Ln 161, Col 54



It's done

THANK YOU

You can find the complete project code on my GitHub Repository

https://github.com/neha071999/SQL_Music_store_analysis