

Aim: To upload raw data containing CSV files to BigQuery and answer key business questions.

CSV files uploaded: Album, artist, customer, employee, genre, invoice, invoice line, media type, playlist, playlist track, track.

Key business questions to answer:

1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are top 3 values of 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.
6. 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.
7. 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.
8. 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.
9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.
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.
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.

Tasks: Querying using JOIN, Common Table and Recursive.

Insights:

1. Senior most employee is Madan Mohan and works as a Senior General Manager.
2. Top 3 countries with most invoices are USA (131), Canada (76), Brazil (61).
3. Top 3 values of total invoice: 23.7599, 19.8, 19.8.
4. Prague has the best customers with total invoices of 273.2399.
5. The person who spent the most money was Frantisek Wichterlova (144.54).
6. A snippet of the long list has been included in the next pages.
7. A snippet of the long list has been included in the next pages.

8. The longest song was Occupation / Precipice with 5286953 milliseconds.
9. A snippet of the long list has been included in the next pages.
10. A snippet of the long list has been included in the next pages.
11. A snippet of the long list has been included in the next pages.

Guided by Rishabh Mishra <https://www.youtube.com/@RishabhMishraOfficial>

My GitHub: <https://github.com/RahulSureshChavan/Guided-Project-SQL-Music-Store-Analysis>

Following are the snippets answering the questions:

Question 1:

The screenshot shows a SQL IDE interface. On the left is an 'Explorer' pane with a search bar and a list of database resources: album2, artist, customer, employee, genre, invoice, invoice_line, media_type, playlist, and playlist_track. The main editor area is titled 'Untitled query' and contains the following SQL code:

```
1 # Who is the senior most employee based on Job Title?
2
3 SELECT
4 *
5 FROM
6 premium-strata-428618-p6.music_store_data.employee
7 ORDER BY levels DESC
8 LIMIT 1
9
10 # Madan Mohan is the senior most employee based on Job Title
```

Below the editor, the 'Query results' section is displayed. It has tabs for 'JOB INFORMATION', 'RESULTS', 'CHART', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'. The 'RESULTS' tab is active, showing a table with the following data:

Row	employee_id	last_name	first_name	title
1	9	Madan	Mohan	Senior General Manager

Question 2:

⏪ ⏴ × 🔍 *Untitled query × 🔍 *Untitled query × 🔍 *Untitled query × ⏵ ⏶ + ▾ ⋮

🔍 Untitled query ▶ RUN + 👤 SHARE ▾ 🕒 SCHEDULE 💾 SAVE ▾ ⬇️ DOWNLOAD ⋮ ✅ This ..

```
1 # Which countries have the most invoices?
2
3 SELECT
4   billing_country,
5   COUNT(*) as Number_of_invoices
6 FROM
7   premium-strata-428618-p6.music_store_data.invoice
8 GROUP BY
9   billing_country
10 ORDER BY
11   Number_of_invoices DESC
12
13 # Top 3 include USA, Canada and Brazil.
```

Press Option+F1 for Accessibility Option

Query results 📄 SAVE RESULTS ▾ 📊 EXPLORE DATA ▾ ⬆️

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	billing_country ▾	Number_of_invoices	
1	USA	131	
2	Canada	76	
3	Brazil	61	
4	France	50	

Results per page: 50 ▾ 1 – 24 of 24 |< < > >|

Question 3:

< 🔍 *Untitled query X 🔍 *Untitled query X 🔍 *Untitled query X > +

🔍 Untitled query ▶ RUN 💾 SAVE ⬇️ DOWNLOAD 👤 SHARE 🕒 SCHEDULE 🟢 This ..

```
1 # What are top 3 values of total invoice?
2
3 SELECT
4   total
5 FROM
6   premium-strata-428618-p6.music_store_data.invoice
7 ORDER BY total DESC
8 LIMIT 3
9
10 # 23.7599, 19.8, 19.8
```

Press Option+F1 for Accessibility Option

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	total
1	23.75999999999...
2	19.8
3	19.8

Question 4:

< 🔍 *Untitled query X 🔍 *Untitled query X 🔍 *Untitled query X 🔍 *Untitled query X 🔍 *Untitled query X > +

🔍 Untitled query ▶ RUN 👤 SHARE 🕒 SCHEDULE ⚙️ MORE 💾 SAVE ⬇️ DOWNLOAD 🟢 This query will pr...

```
1 # Which city has the best customers? Output: One city and its total invoices.
2
3 SELECT
4   billing_city,
5   SUM(total) as total_invoices
6 FROM
7   premium-strata-428618-p6.music_store_data.invoice
8 GROUP BY
9   billing_city
10 ORDER BY
11   total_invoices DESC
12 LIMIT 1
13
14 # Prague with 273.2399 total invoices.
```

Press Option+F1 for Accessibility Option

Query results 📄 SAVE RESULTS 📊 EXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	billing_city	total_invoices
1	Prague	273.2399999999...

Question 5:

< *Untitled query × 🔍 *Untitled query × 🔍 *Untitled query × 🔍 *Untitled query × 🔍 *Untitled query > +

Untitled query ▶ RUN 🕒 SCHEDULE ⚙️ MORE 💾 SAVE ⬇️ DOWNLOAD 👤 SHARE ✅ This query will pr...

```
1 # Best customer: The person who spent the most money.
2
3 SELECT
4   c.customer_id,
5   c.first_name,
6   c.last_name,
7   SUM(i.total) as total
8 FROM
9   premium-strata-428618-p6.music_store_data.customer as c
10  JOIN premium-strata-428618-p6.music_store_data.invoice as i
11    ON c.customer_id = i.customer_id
12
13
14
15
16 ORDER BY total DESC
17 LIMIT 1
18
19 # František Wichterlová spent 144.54 Dollars.
20
```

in current tab
- open in new tab,
+ Tap - open in split tab)

Press Option+F1 for Accessibility Option

Query results

📄 SAVE RESULTS 📊 EXPLORE DATA ⌵

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_id	first_name	last_name	total		
1	5	František	Wichterlová	144.5400000000...		

Question 6:

Untitled query

RunShareScheduleMoreSaveDownload

```
1 # Write a query to return the email, first name, last name, and genre of all Rock Music listeners. Return the list ordered alphabetically by email starting with A.
2
3 SELECT DISTINCT
4   email,
5   first_name,
6   last_name
7 FROM
8   premium-strata-428618-p6.music_store_data.customer as c
9 JOIN premium-strata-428618-p6.music_store_data.invoice as i
10  ON c.customer_id = i.customer_id
11 JOIN premium-strata-428618-p6.music_store_data.invoice_line as il
12  ON i.invoice_id = il.invoice_id
13 WHERE
14   track_id IN(
15     SELECT
16       track_id
17     FROM
18       premium-strata-428618-p6.music_store_data.track as t
19     JOIN premium-strata-428618-p6.music_store_data.genre as g
20      ON t.genre_id = g.genre_id
21     WHERE
22       g.name = 'Rock'
23   )
24 ORDER BY c.email
```

Press Option+F1 for Accessibility Options

Query results

Save ResultsExplore Data

Job InformationResultsChartJSONExecution DetailsExecution Graph

Row	email	first_name	last_name
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjorn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	dave.bernard@yahoo.fr	Dave	Bernard

Results per page: 501 - 50 of 59

Question 7:

⏮

⏪

🔍 *Untitled query

✕

🔍 *Untitled query

✕

🔍 *Untitled query

✕

🔍 *Untitled query

✕

🔍 *Un

➤

+

⋮

🔍 Untitled query

▶ RUN

💾 SAVE

⬇️ DOWNLOAD

👤 SHARE

🕒 SCHEDULE

⚙️ MORE

1

Write a query that returns the Artist name and total track count of the top 10 rock bands.

2

3

SELECT

4

· a.artist_id,

5

· a.name,

6

· COUNT(a.artist_id) as number_of_songs

7

FROM premium-strata-428618-p6.music_store_data.artist as a

8

JOIN premium-strata-428618-p6.music_store_data.album as al

9

ON a.artist_id = al.artist_id

10

JOIN premium-strata-428618-p6.music_store_data.track as t

11

ON al.album_id = t.album_id

12

JOIN premium-strata-428618-p6.music_store_data.genre as g

13

ON t.genre_id = g.genre_id

14

WHERE

15

· g.name = 'Rock'

16

GROUP BY a.artist_id, a.name

17

ORDER BY number_of_songs DESC

Press Option+F1 for Accessibility Options

Query results

📄 SAVE RESULTS

📊 EXPLORE DATA

⌵

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	artist_id	name	number_of_songs
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81

Results per page: 50 1 – 50 of 51 ⏮ ⏪ ⏩ ⏭

Question 8:

<

query

×

🔍

*Untitled query

×

🔍

*Untitled query

×

🔍

*Untitled query

×

🔍

>

+

⋮

xplorer

Untitled query

▶

RUN

📄

SAVE

▼

⬇

DOWNLOAD

👤

SHARE

▼

🕒

SCHEDULE

⋮

✔

This ..

1

Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

2

3

SELECT

4

· · name,

5

· · milliseconds

6

FROM

7

· · premium-strata-428618-p6.music_store_data.track

8

· WHERE milliseconds > (

9

· · · SELECT AVG(milliseconds)

10

· · · FROM premium-strata-428618-p6.music_store_data.track

11

·)

12

· ORDER BY milliseconds DESC

13

14

Occupation / Precipice with 5286953 milliseconds.

Press Option+F1 for Accessibility Option

Query results

📄

SAVE RESULTS

▼

📊

EXPLORE DATA

▼

⬆

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	name	milliseconds
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838

Results per page: 50 1 – 50 of 494 |< < > >|

Question 9:

Untitled query

RUNSHARESCHEDULEMORESAVEDOWNLOAD

```
1 # Find how much money spent by each customer on artists? Write a query to return customer name, artist name and total spent.
2
3 SELECT
4   c.customer_id,
5   c.first_name,
6   c.last_name,
7   ar.name,
8   SUM(il.unit_price*il.quantity) as amount_spent
9 FROM
10  premium-strata-428618-p6.music_store_data.customer as c
11 JOIN premium-strata-428618-p6.music_store_data.invoice as i
12 ON c.customer_id = i.customer_id
13 JOIN premium-strata-428618-p6.music_store_data.invoice_line as il
14 ON i.invoice_id = il.invoice_id
15 JOIN premium-strata-428618-p6.music_store_data.track as t
16 ON il.track_id = t.track_id
17 JOIN premium-strata-428618-p6.music_store_data.album as a
18 ON t.album_id = a.album_id
19 JOIN premium-strata-428618-p6.music_store_data.artist as ar
20 ON a.artist_id = ar.artist_id
21 GROUP BY
22   c.customer_id,
23   c.first_name,
24   c.last_name,
25   ar.name
26 ORDER BY amount_spent DESC
27
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_id	first_name	last_name	name	amount_spent	
1	46	Hugh	O'Reilly	Queen	27.71999999999...	
2	42	Wyatt	Girard	Frank Sinatra	23.75999999999...	
3	3	François	Tremblay	The Who	19.79999999999...	
4	29	Robert	Brown	Creedence Clearwater Revival	19.79999999999...	
5	5	František	Wichterlová	Kiss	19.79999999999...	
6	32	Aaron	Mitchell	James Brown	19.79999999999...	
7	6	Helena	Holý	Red Hot Chili Peppers	19.79999999999...	
8	38	Niklas	Schröder	Queen	18.81	

Load more

Question 10:

Untitled query

RUN

SHARE

SCHEDULE

MORE

SAVE

DOWNLOAD

This query will process 177.23 KB when run.

```
1 # Find the most popular genre for each country. Popular genre is the one with highest purchases.
2
3 # Using Common Table
4
5 WITH popular_genre AS
6 (
7     SELECT
8         COUNT(il.quantity) AS purchases, c.country, g.name, g.genre_id, ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity)DESC) AS RowNo
9     FROM premium-strata-428618-p6.music_store_data.invoice_line AS il
10    JOIN premium-strata-428618-p6.music_store_data.invoice AS i
11    ON i.invoice_id = il.invoice_id
12    JOIN premium-strata-428618-p6.music_store_data.customer AS c
13    ON c.customer_id = i.customer_id
14    JOIN premium-strata-428618-p6.music_store_data.track AS t
15    ON t.track_id = il.track_id
16    JOIN premium-strata-428618-p6.music_store_data.genre AS g
17    ON g.genre_id = t.genre_id
18    GROUP BY c.country, g.name, g.genre_id
19    ORDER BY c.country ASC, purchases DESC
20 )
21 SELECT *
22 FROM popular_genre
23 WHERE RowNo <= 12
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS		EXECUTION GRAPH	
Row		purchases	country	name	genre_id	RowNo		
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		
11		211	France	Rock	1	1		
12		194	Germany	Rock	1	1		

Load more

Results per page: 501 – 24 of 24

```

1 # Find the most popular genre for each country. Popular genre is the one with highest purchases.
2
3 # Using Recursive
4
5 WITH RECURSIVE
6 sales_per_country AS(
7     SELECT COUNT(*) AS purchases_per_genre, c.country, g.name, g.genre_id
8     FROM premium-strata-428618-p6.music_store_data.invoice_line AS il
9     JOIN premium-strata-428618-p6.music_store_data.invoice AS i
10    ON i.invoice_id = il.invoice_id
11    JOIN premium-strata-428618-p6.music_store_data.customer AS c
12    ON c.customer_id = i.customer_id
13    JOIN premium-strata-428618-p6.music_store_data.track AS t
14    ON t.track_id = il.track_id
15    JOIN premium-strata-428618-p6.music_store_data.genre AS g
16    ON g.genre_id = t.genre_id
17    GROUP BY 2,3,4
18    ORDER BY 2
19 ),
20 max_genre_per_country AS (SELECT MAX(purchases_per_genre) AS max_genre_number, country
21 FROM sales_per_country
22 GROUP BY 2
23 ORDER BY 2)
24
25 SELECT sales_per_country.*
26 FROM sales_per_country
27 JOIN max_genre_per_country ON sales_per_country.country = max_genre_per_country.country
28 WHERE sales_per_country.purchases_per_genre = max_genre_per_country.max_genre_number;
29

```

Press Alt+F1 for Accessibility Option

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	purchases_per_genre	country		name	genre_id	
1	561	USA		Rock	1	
2	333	Canada		Rock	1	
3	211	France		Rock	1	
4	40	Poland		Rock	1	
5	102	India		Rock	1	
6	194	Germany		Rock	1	
7	108	Portugal		Rock	1	
8	46	Finland		Rock	1	
9	205	Brazil		Rock	1	

Results per page: 50 1 - 24 of 24

Job history

REFRESH

Question 11:

Untitled query

RUN

SHARE

SCHEDULE

MORE

SAVE

DOWNLOAD

```
1 # 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
2 top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.
3 WITH Customer_with_country AS (
4     SELECT c.customer_id,c.first_name,c.last_name,i.billing_country,SUM(i.total) AS total_spending,
5           ROW_NUMBER() OVER(PARTITION BY i.billing_country ORDER BY SUM(i.total) DESC) AS RowNo
6     FROM premium-strata-428618-p6.music_store_data.invoice AS i
7     JOIN premium-strata-428618-p6.music_store_data.customer AS c
8     ON c.customer_id = i.customer_id
9     GROUP BY i.billing_country,2,3,4
10    ORDER BY 4 ASC,5 DESC)
11 SELECT * FROM Customer_with_country WHERE RowNo <= 1
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	customer_id	first_name	last_name	billing_country	total_spending	RowNo
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	108.89999999999999	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000000	1
8	5	František	Wichterlová	Czech Republic	144.54000000000000	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.05000000000000	1

Load more

Results were stored in a separate dataset:

Results

Best_customer

Best_customer_by_...

Country_TopCustom...

Countrywise_popular...

Longest_songs

Money_spent_by_ea...

Number_of_invoices

Rock_artists_with_m...

Rock_listeners

Senior

Top_3_values_invoice