Analyze Data for An Internet Music Outlet With SQL

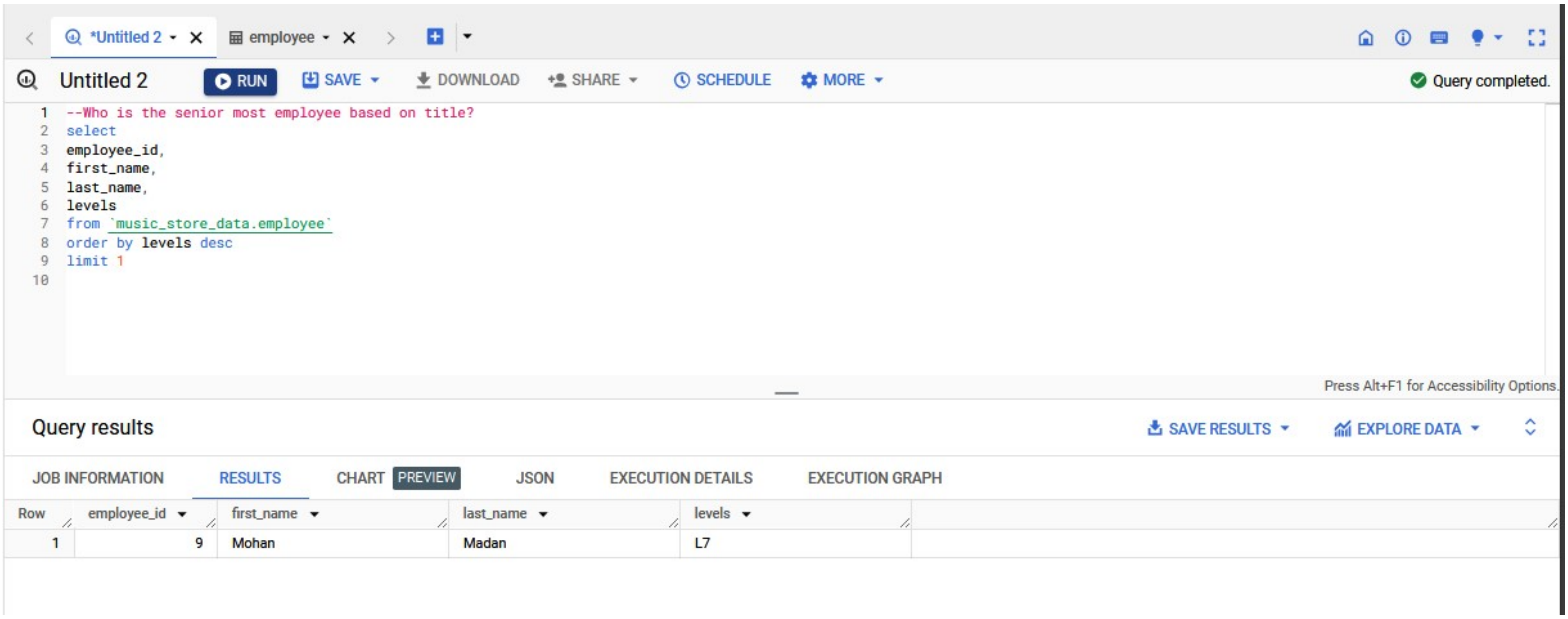
In this project I did data analysis for an online music store. Some of the key ways I could make a **significant impact** on the business was by answering these questions:

1. **Targeted Marketing Strategies**: Utilize the insights from the analysis to tailor marketing efforts.
2. **Resource Allocation Optimization**: Allocate resources efficiently by understanding customer behavior.
3. **Revenue Maximization**: Leverage the analysis to boost revenue.
4. **Customer Retention and Acquisition**: Enhance customer relationships.
5. **Strategic Decision-Making**: Provide actionable insights to management.

⟹ All the queries, their solutions and explanations are given below.

Query 1:
Finding the senior most employee form the employee table.

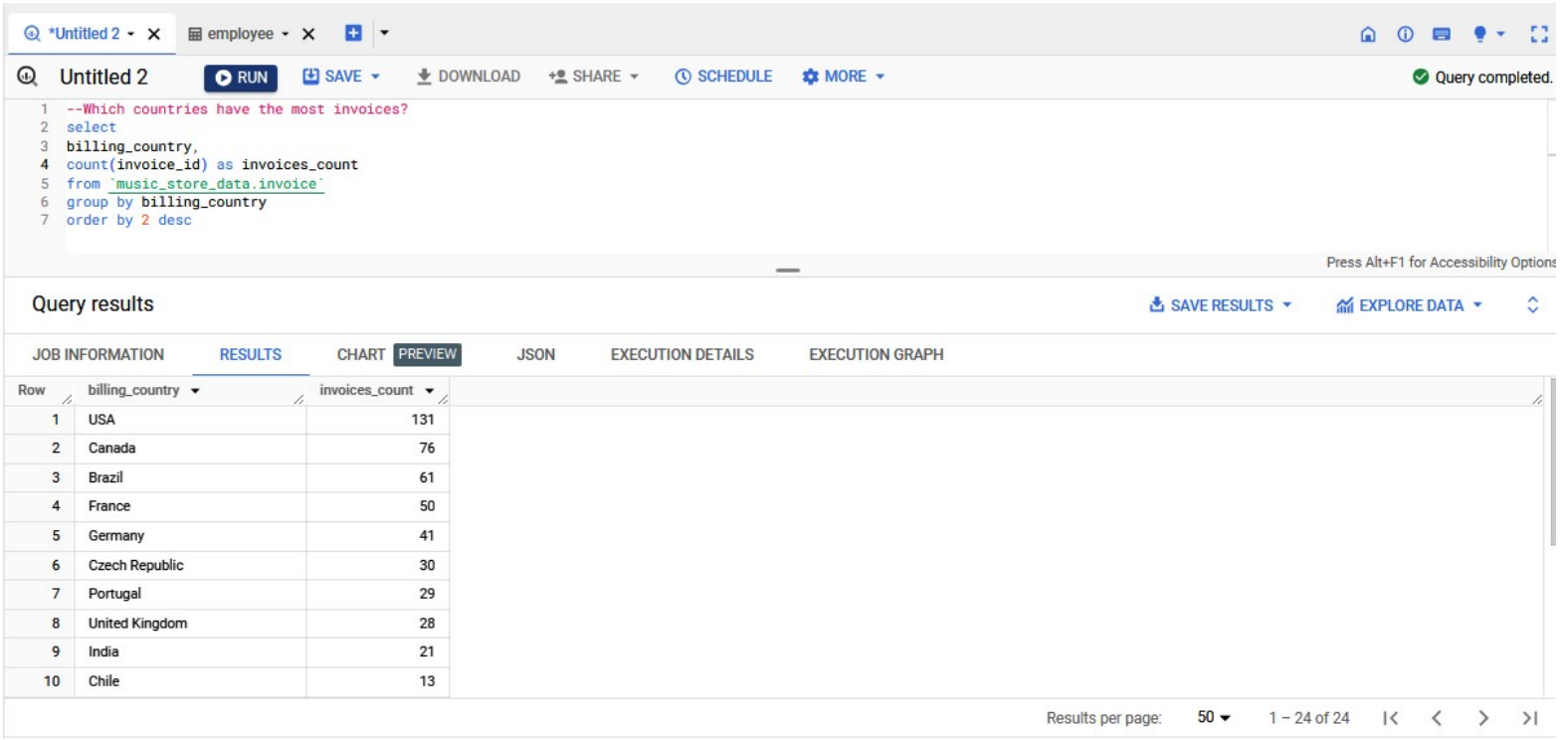
```sql
Who is the senior-most employee based on title?
SELECT
employee_id,
first_name,
last_name,
levels
FROM "music_store_data.employee"
ORDER BY levels DESC
LIMIT 1;
```



Query 2:

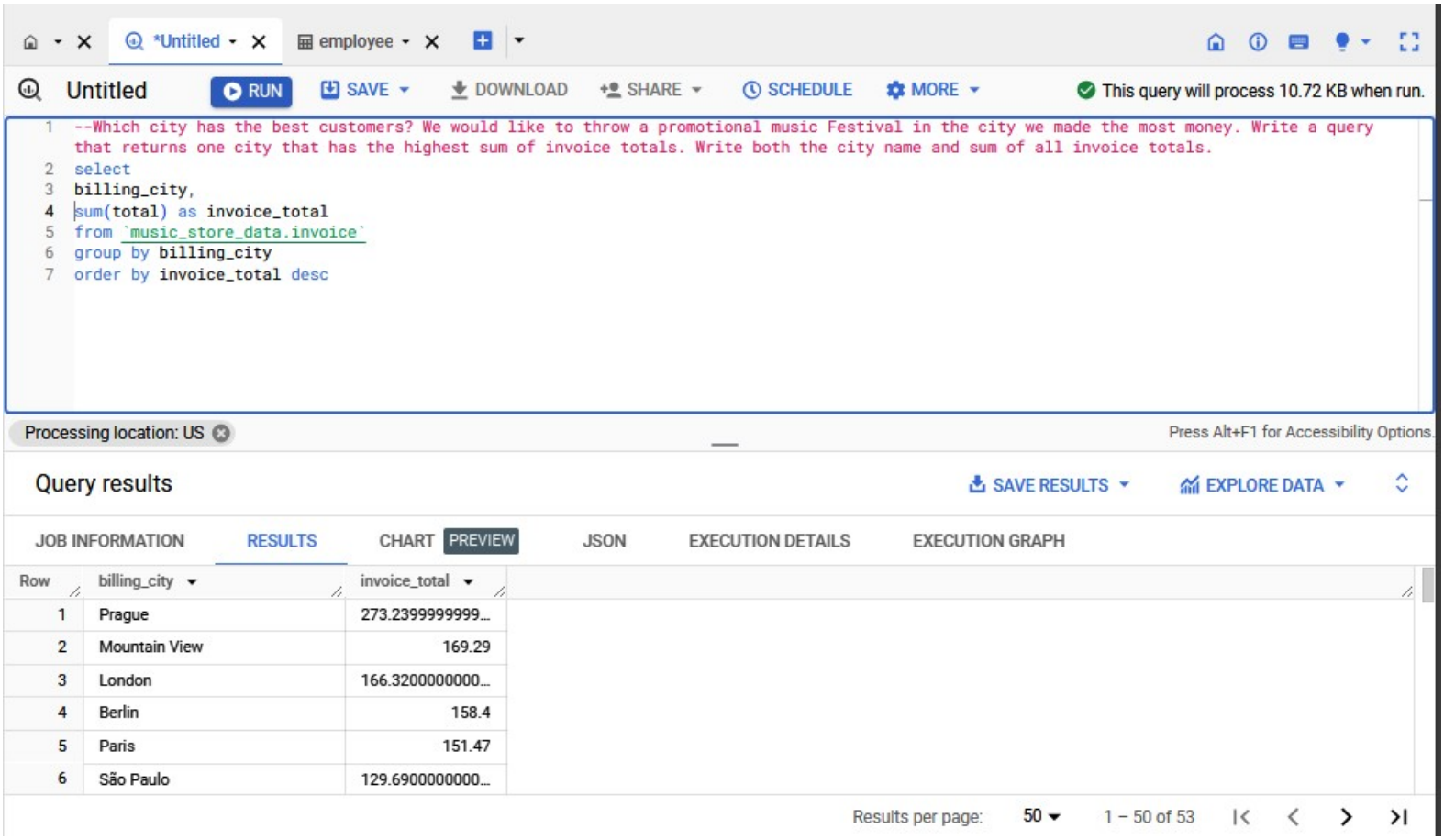
Finding top countries by the number of invoices by grouping with “billing country”.

```sql
--Which countries have the most invoices?
select
billing_country,
count(invoice_id) as invoices_count
from 'music_store_data.invoice'
group by billing_country
order by desc;
```

Query 3:
Finding the city with top sales. Here we take the sum of “total” column and group the data by the “billing city”.

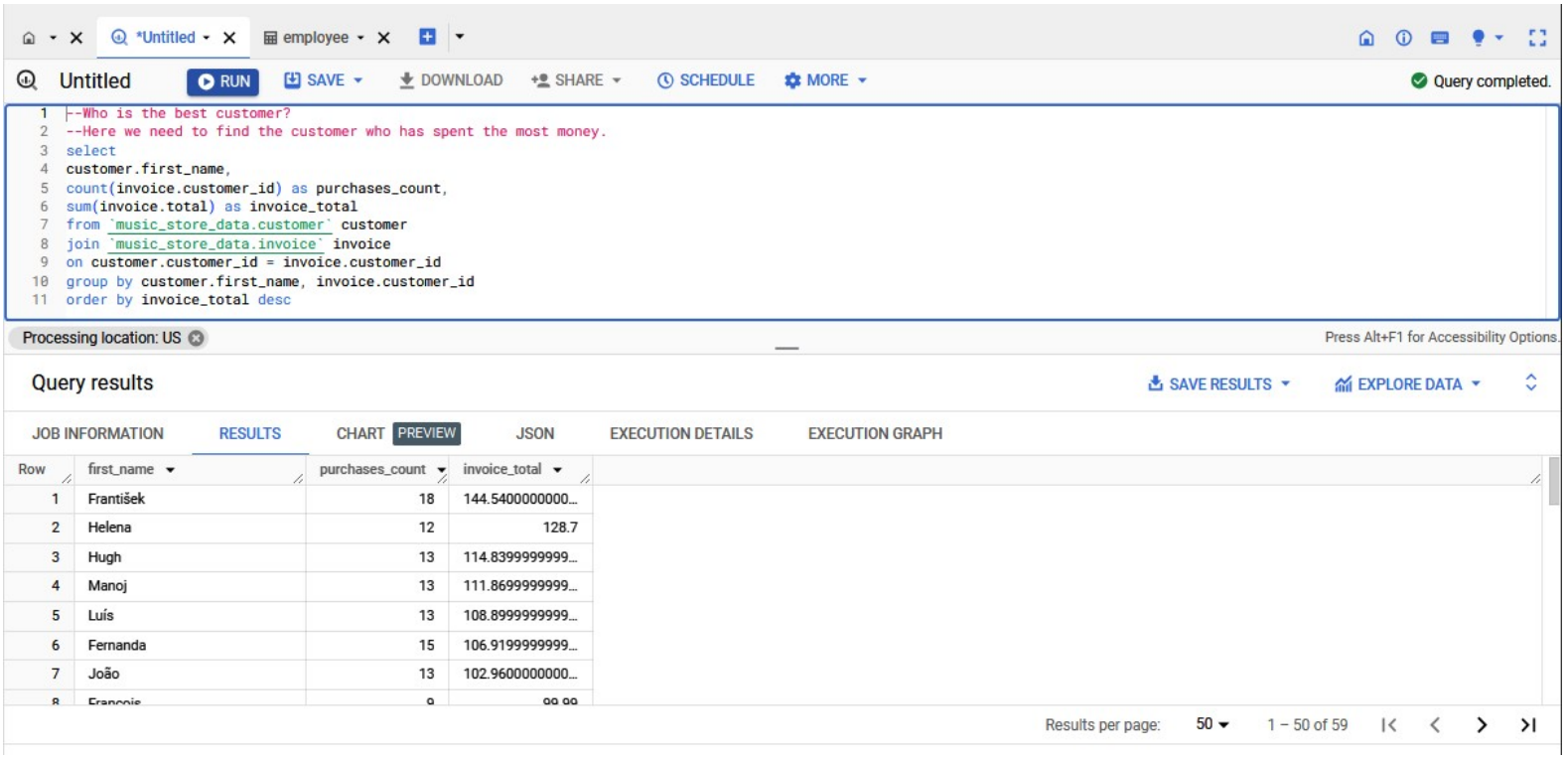
```
--which city has the best customers? We would like to throw a promotional music festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Write both the city name and sum of all invoice totals.
select
billing_city,
sum(total) as invoice_total
from 'music_store_data_invoice'
group by billing_city
order by invoice_total desc
limit 1;
```

Query 4:

Finding out the best customer for our company. Since customer data is present in the “Customers” table and their purchasing data is in the “Invoice” table, we perform a join operation to obtain the required results.

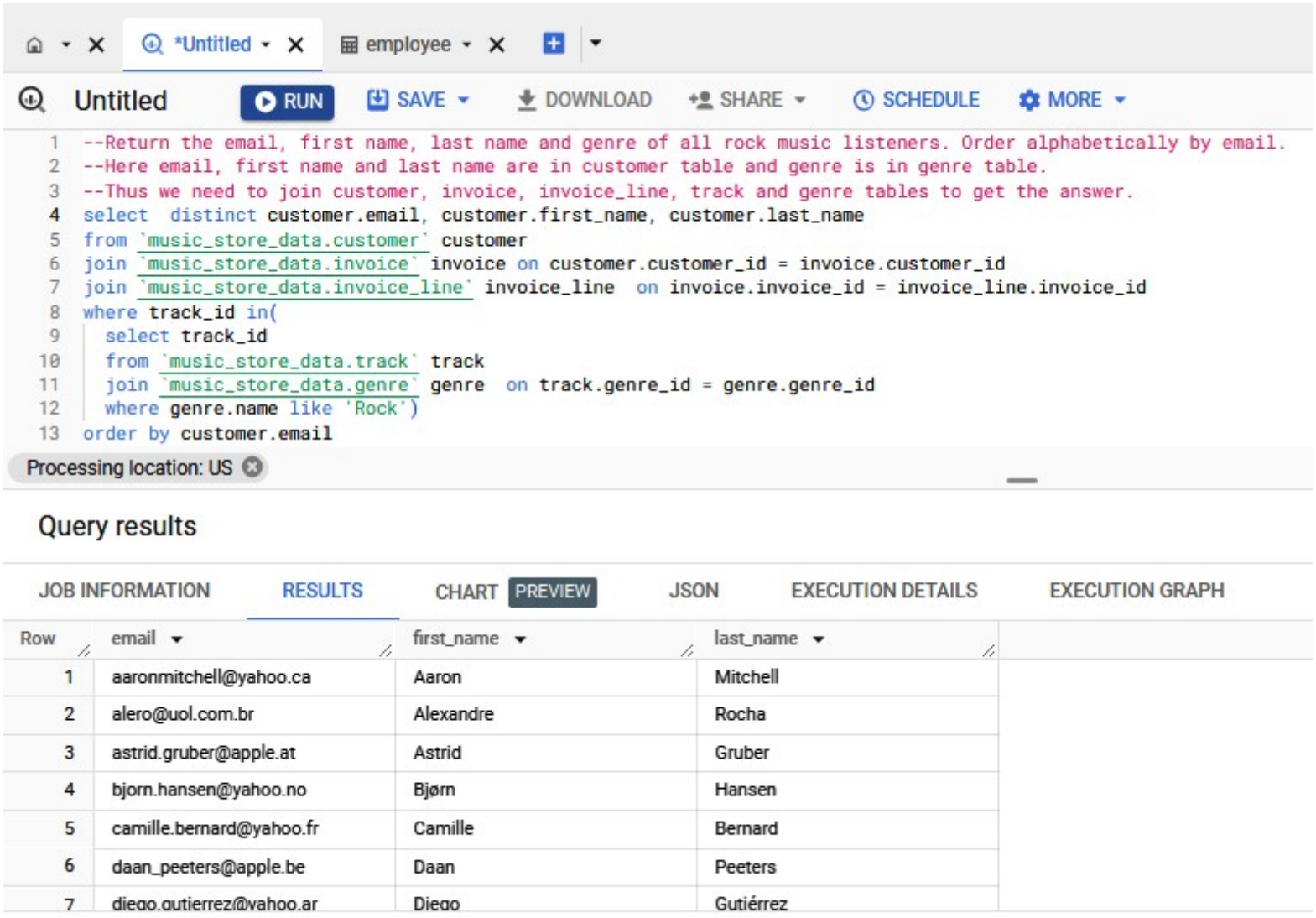

```sql
--Who is the best customer?
--Here we need to find the customer who has spent the most money.
select
customer.first_name,
count(invoice.customer_id) as purchase_count,
sum(invoice.total) as invoice_total
from 'music_store_data.customer' customer
join 'music_store_data.invoice' invoice
on customer.customer_id = invoice.customer_id
group by customer.first_name, invoice.customer_id
order by invoice_total desc;
```



Query 5:

See what the data for rock music listeners looks like. Since email, first name and last name are in the "Customer" table and genre is in the "Genre" table, we will join them. But these table can't be joined directly as they share a relation. Thus, we join these three tables: Customer, Invoice and Genre tables.

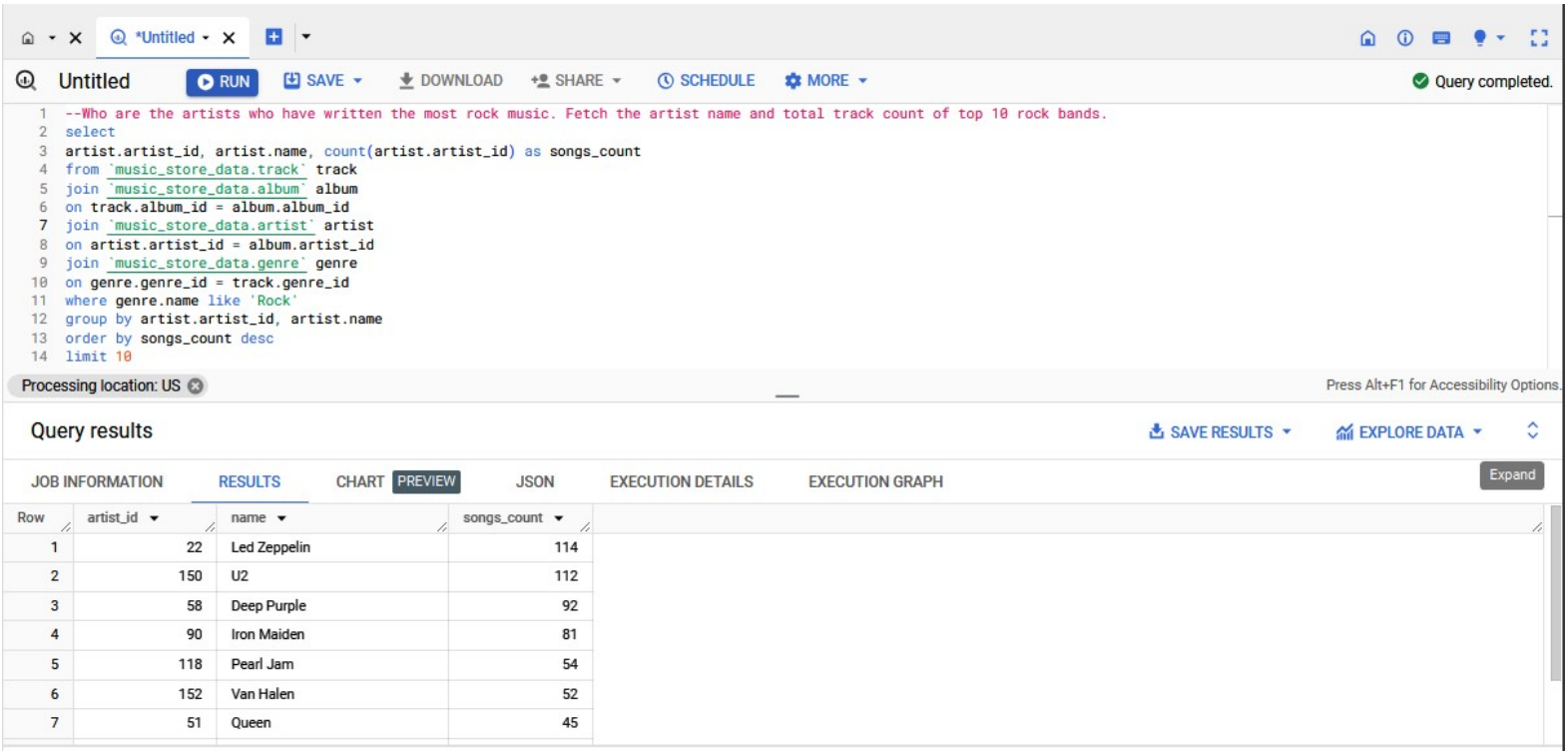

```
1. Return the email, first name, last name and genre of all rock music
   listeners. Order alphabetically my email.
2. Here email, first name, last name are in cudstomer table and genre is in
   genre table.
3. Thus we need to join customer, invoice, invoice_line, track and genre tables
   to get the answer.
4. select distinct customer.email, customer.first_name, customer.last_name
5. from "music_store_data.customer" customer
6. join "music_store_data.invoice" invoice on customer.customer_id =
   invoice.customer_id
7. join "music_store_data.invoice_line" invoice_line on invoice.invoice_id =
   invoice_line.invoice_id
8. where track_id in(
       select track_id
       from "music_store_data.track" track
       join "music_store_data.genre" genre on track.genre_id = genre.genre_id
       where genre.name like 'rock')
9. order by customer.email;
```



Query 6:

Finding the Rock band popularity based on track count. By looking at the schema we realise that we need to join 4 tables in order to obtain the desired result. Finally group the data by artist id and name to obtain the top artists.

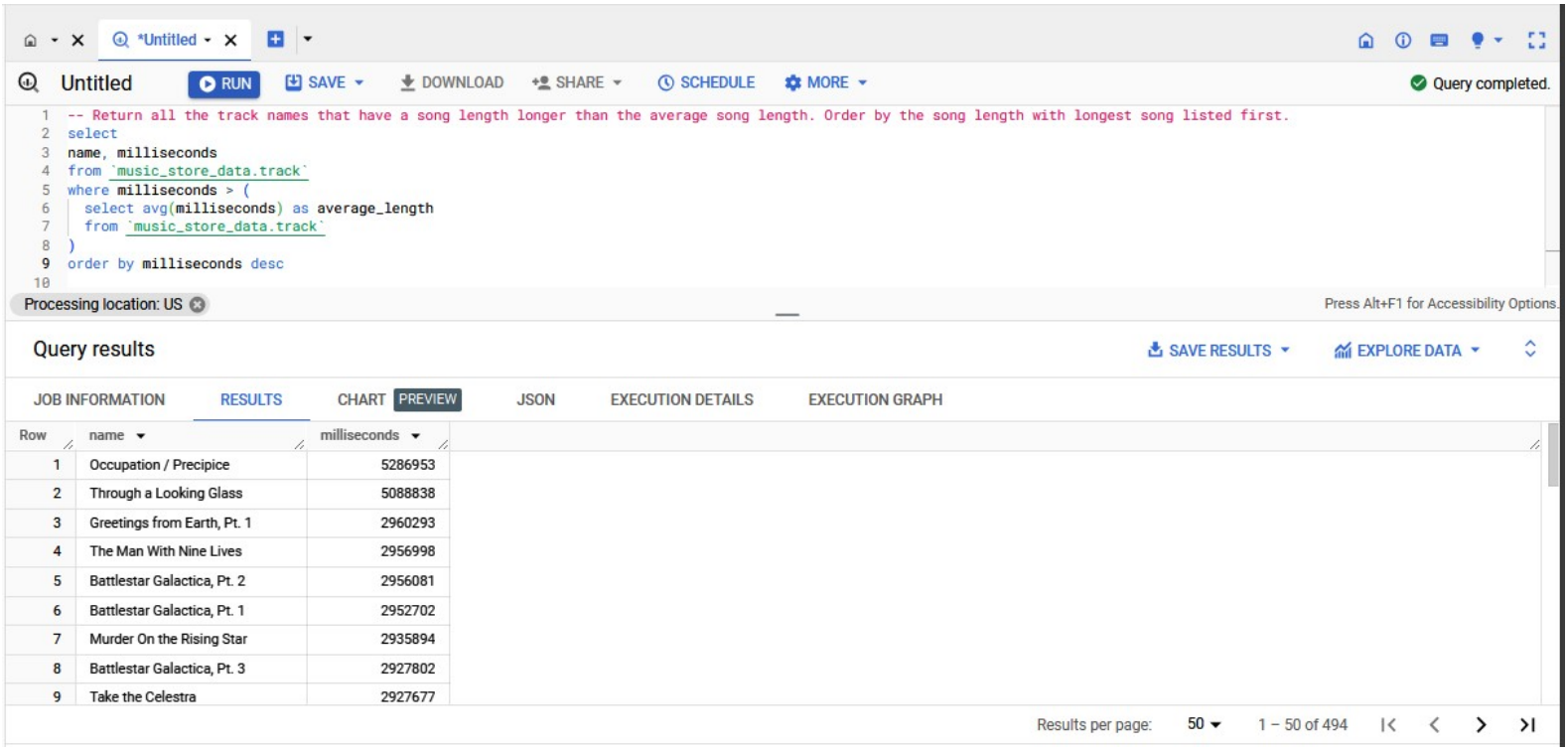

```
-- Who are the artists who have written the most rock music? Fetch the artist name and total track count of top 10 rock bands.
select
artist.artist_id, artist.name, count(artist.artist_id) as songs_count
from 'music_store_data.track' track
join 'music_store_data.album' album
on track.album_id = album.album_id
join 'music_store_data.artist' artist
on artist.artist_id = album.artist_id
join 'music_store_data.genre' genre
on genre.genre_id = track.genre_id
where genre.genre_title = 'Rock'
group by artist.artist_id, artist.name
order by songs_count desc
limit 10;
```



Query 7:

Finding track names that have song length longer than average song length. To retrieve all track names with a song length longer than the average, we'll need to calculate the average song length first. Then we can compare each track's length to this average. Analysing the longest tracks provides valuable insights for content curation, user engagement, and revenue optimization.


```sql
-- Return all the track names that have a song length longer than the average song length. Order by 'the song' length with 'longest' song listed first.
select
name, milliseconds
from 'music_store_data.track'
where milliseconds > (
    select avg(milliseconds) as average_length
    from 'music_store_data.track')
order by milliseconds desc
limit 10;
```



Query 8:

Finding how much each customer has spent on artists. We combine data from multiple tables (e.g., `customer`, `invoice`, `invoice_line`, `track`, `album`, and `artist`) to calculate the total amount spent by each customer on artists. This query provides valuable insights for revenue growth and customer engagement.


```
1. --Find how much amount was spent by each customer on artists? Write a query
   to return customer name, artist name and total spent

2. WITH best_selling_artist AS (
3. SELECT artist.artist_id, artist.name AS artist_name,
   SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
4. FROM 'music_store_data.invoice_line' invoice_line
      JOIN 'music_store_data.track' track
      ON track.track_id = invoice_line.track_id
      JOIN music_store_data.album album
      ON album.album_id = track.album_id
      JOIN music_store_data.artist artist
      ON artist.artist_id = album.artist_id
      GROUP BY 1,2
      ORDER BY 3 DESC
5. LIMIT 1
6. )
7. SELECT customer.customer_id,
   customer.first_name,customer.last_name,bsa.artist_name,
   SUM(invoice_line.unit_price*invoice_line.quantity) as amount_spent
8. FROM 'music_store_data.invoice' invoice
9. JOIN 'music_store_data.customer' customer
10. ON customer.customer_id = invoice.customer_id
11. JOIN 'music_store_data.invoice_line' invoice_line
12. ON invoice_line.invoice_id = invoice.invoice_id
13. JOIN 'music_store_data.track' track
14. ON track.track_id = invoice_line.track_id
15. JOIN 'music_store_data.album' album
16. ON album.album_id = track.album_id
17. JOIN best_selling_artist bsa
18. ON bsa.artist_id=album.artist_id
19. GROUP BY 1,2,3,4
20. ORDER BY 5 DESC;
```

*Untitled

Untitled RUN SAVE DOWNLOAD SHARE SCHEDULE MORE Query completed.

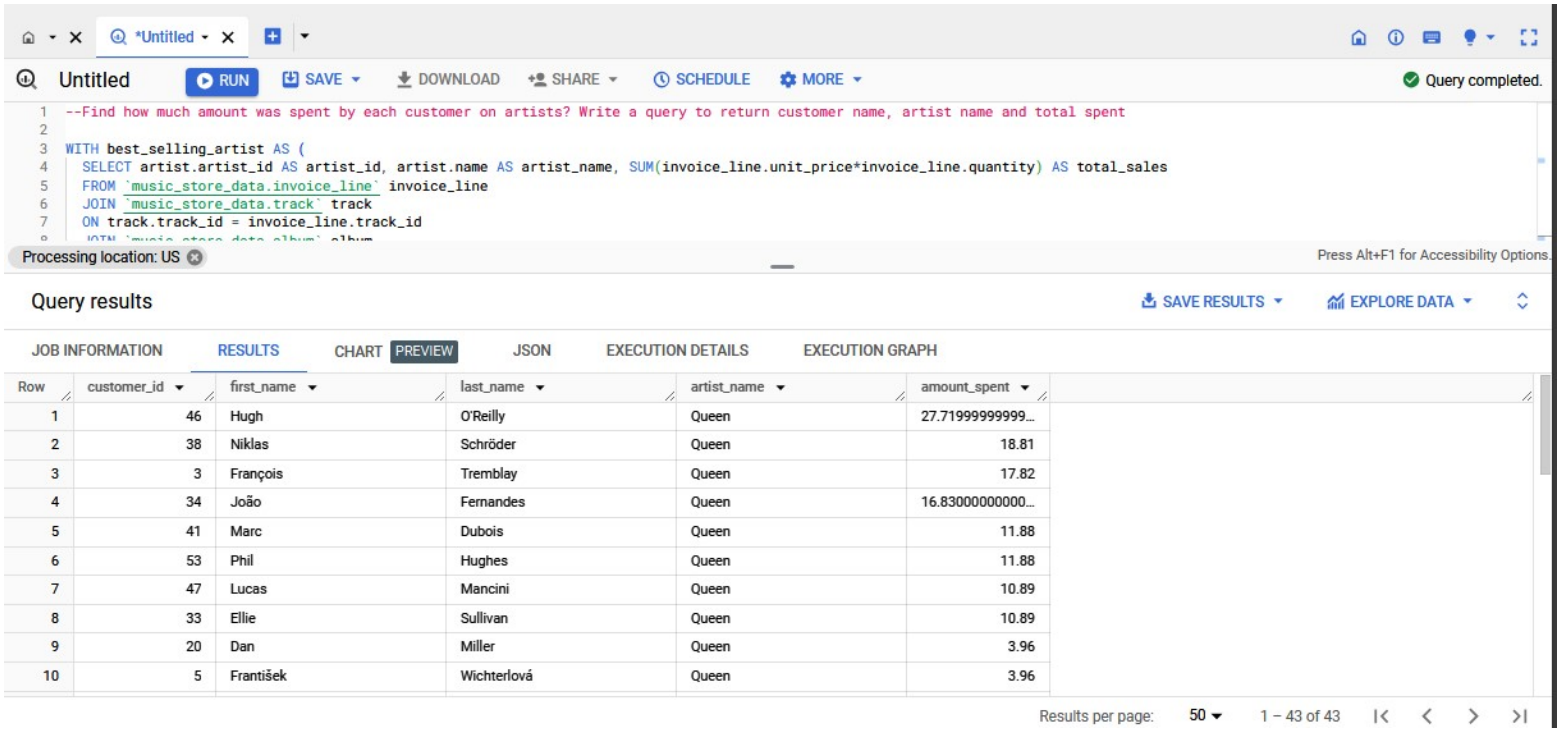
```sql
--Find how much amount was spent by each customer on artists? Write a query to return customer name, artist name and total spent

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 `music_store_data.invoice_line` invoice_line
  JOIN `music_store_data.track` track
  ON track.track_id = invoice_line.track_id
  JOIN `music_store_data.album` album
  ON album.album_id = track.album_id
  JOIN `music_store_data.artist` artist
  ON artist.artist_id = album.artist_id
  GROUP BY 1,2
  ORDER BY 3 DESC
  LIMIT 1
)
SELECT customer.customer_id, customer.first_name, customer.last_name, bsa.artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS amount_spent
FROM `music_store_data.invoice` invoice
JOIN `music_store_data.customer` customer
ON customer.customer_id = invoice.customer_id
JOIN `music_store_data.invoice_line` invoice_line
ON invoice_line.invoice_id = invoice.invoice_id
JOIN `music_store_data.track` track
ON track.track_id = invoice_line.track_id
JOIN `music_store_data.album` album
ON album.album_id = track.album_id
JOIN best_selling_artist bsa
ON bsa.artist_id = album.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC
```

Processing location: US

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

Results image:



Query 9:

Finding the most popular music genre for each country. With the help of a CTE, we calculate the total number of purchases for each genre in each country. The ROW_NUMBER() function assigns a rank to each genre within a country based on the purchase count. PARTITION BY CountryId groups the data by country. Within each country, the ranking is calculated. This analysis provides actionable insights for marketing, content curation, and revenue growth within the music industry.


```
1. --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.

2. with popular_genre as (
3. select count(invoice_line.quantity) as purchases, customer.country,
   genre.name, genre.genre_id,
4. row_number() over(partition by customer.country order by
   count(invoice_line.quantity) desc) as rowno
5. from 'music_store_data.invoice_line' invoice_line
6. join 'music_store_data.invoice' invoice
7. on invoice.invoice_id = invoice_line.invoice_id
8. join 'music_store_data.customer' customer
9. on customer.customer_id = invoice.customer_id
10. join 'music_store_data.track' track
11. on track.track_id = invoice_line.track_id
12. join 'music_store_data.genre' genre
13. on genre.genre_id = track.genre_id
14. group by 2, 3, 4
15. order by 2 asc, 1 desc
16. )
17. select * from popular_genre where rowno <= 1;
```

*Untitled

Untitled

RUN SAVE DOWNLOAD SHARE SCHEDULE MORE

Query completed.

```sql
--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 `music_store_data.invoice_line` invoice_line
  join `music_store_data.invoice` invoice
  on invoice.invoice_id = invoice_line.invoice_id
  join `music_store_data.customer` customer
  on customer.customer_id = invoice.customer_id
  join `music_store_data.track` track
  on track.track_id = invoice_line.track_id
  join `music_store_data.genre` 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
```

Processing location: US

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	purchases	country	name	genre_id	rowno
1	17	Argentina	Alternative & Punk	4	1

Results per page: 50 1 – 24 of 24

Results image:

*Untitled

Untitled

RUN SAVE DOWNLOAD SHARE SCHEDULE MORE

Query completed.

```sql
--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
```

Processing location: US

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

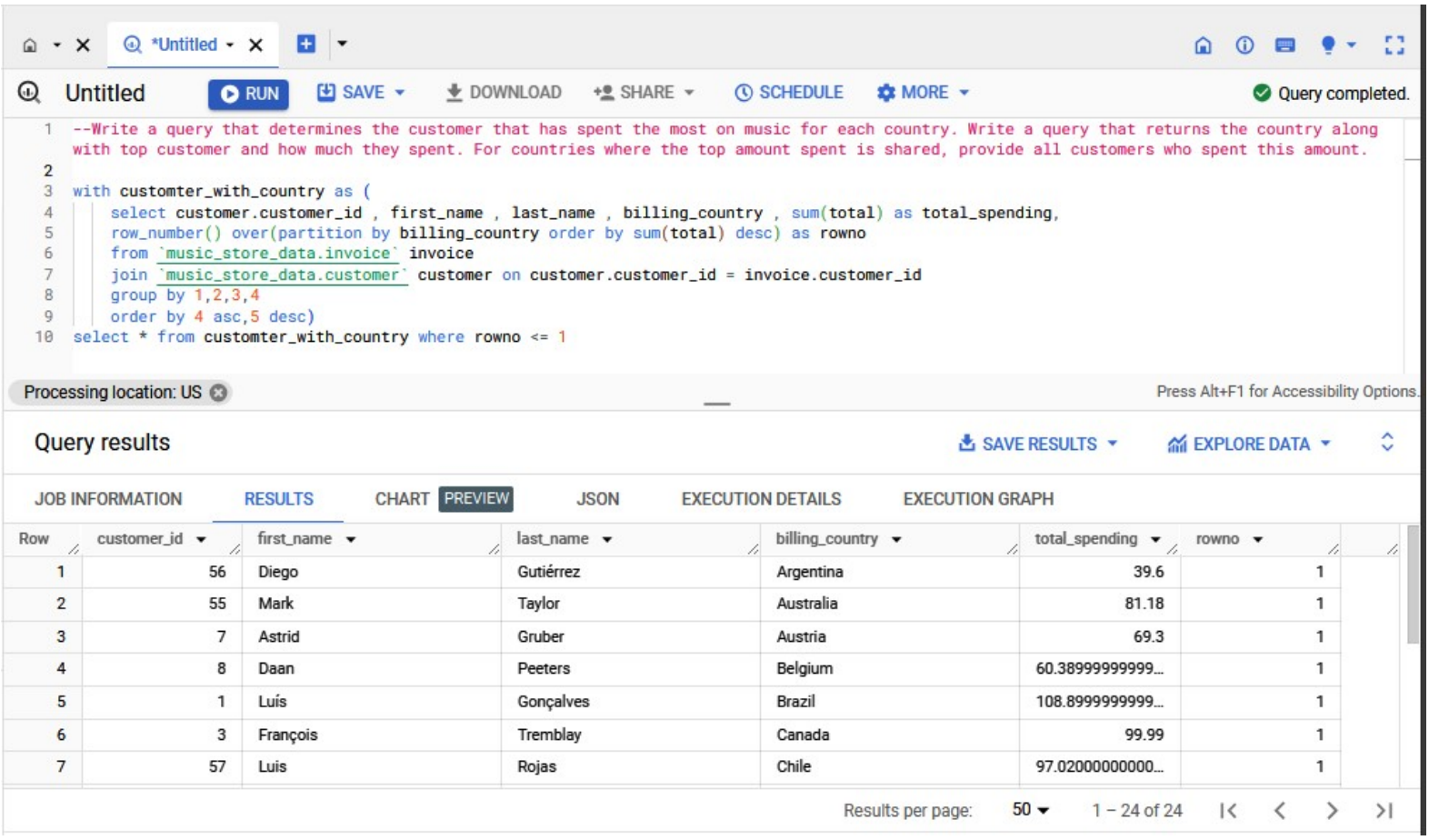
Row	purchases	country	name	genre_id	rowno
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1

Results per page: 50 1 – 24 of 24

Query 10:

Finding the top spending customer for each country based on their spending on music. With the help of a CTE, we calculate the total spending for each customer in each country. Then we assign a row number to each customer within their country, ordered by total spending (descending). Finally, we select rows from the CTE where the row number is 1 (i.e., the top spending customer for each country).


```
--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 top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.
with customer_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 'music_store_data.invoice' invoice
join 'music_store_data.customer' customer on customer.customer_id =
invoice.customer_id
group by 1,2,3,4
order by 4 asc,5 desc)
select * from customer_with_country where rowno <= 1;
```



Results image: