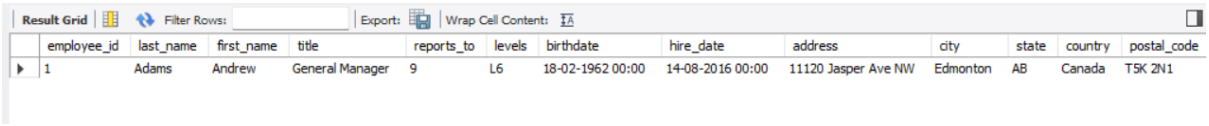
SQL Project- Music Store Data Analysis

Question Set 1 (Easy)

Que1. Who is the senior most employee based on job title?

```sql
-- Who is the most senior employee in the organisation
SELECT * FROM employee
ORDER BY levels desc
LIMIT 1;
```

Output:

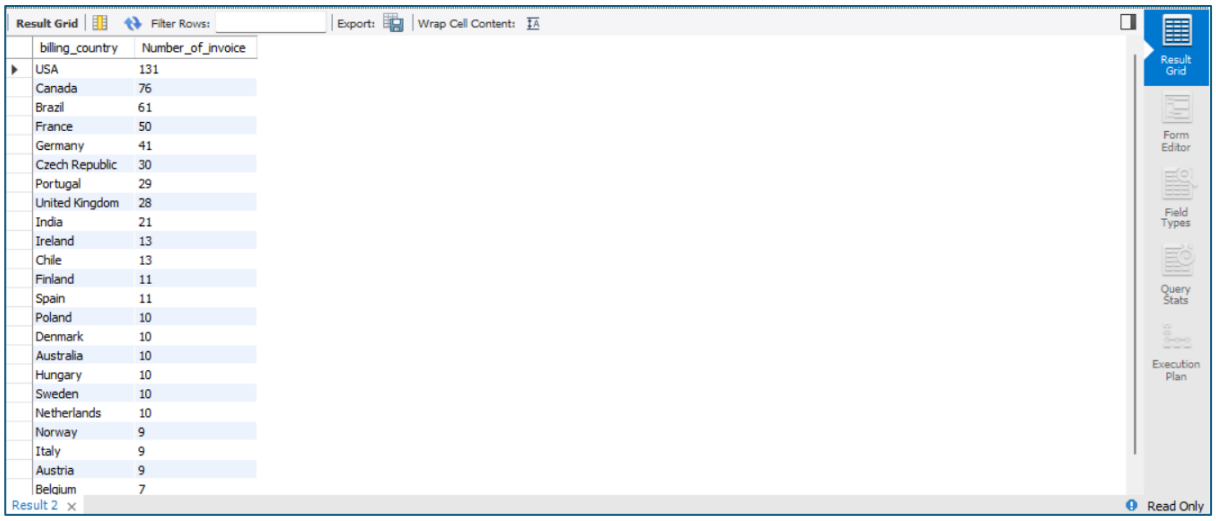


employee_id	last_name	first_name	title	reports_to	levels	birthdate	hire_date	address	city	state	country	postal_code
1	Adams	Andrew	General Manager	9	L6	18-02-1962 00:00	14-08-2016 00:00	11120 Jasper Ave NW	Edmonton	AB	Canada	T5K 2N1

Que2. Which countries have the most Invoices?

```sql
-- Which country has the most billing invoices
SELECT billing_country,count(invoice_id)AS Number_of_invoice FROM invoice
GROUP BY billing_country
ORDER BY count(invoice_id)DESC;
```

Output:

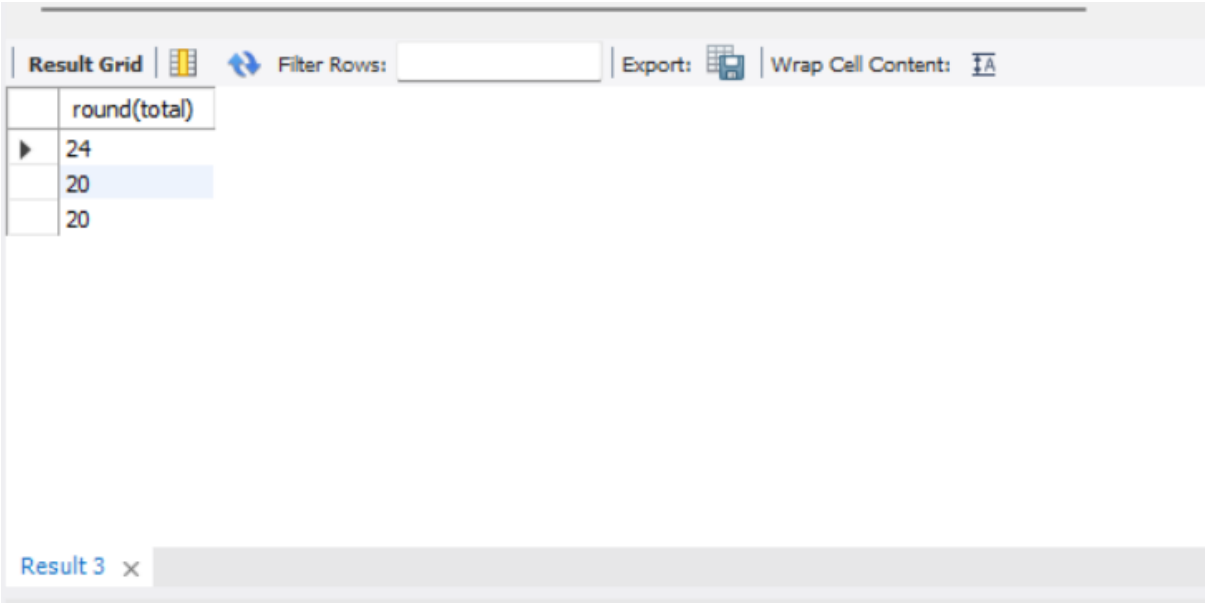


billing_country	Number_of_invoice
USA	131
Canada	76
Brazil	61
France	50
Germany	41
Czech Republic	30
Portugal	29
United Kingdom	28
India	21
Ireland	13
Chile	13
Finland	11
Spain	11
Poland	10
Denmark	10
Australia	10
Hungary	10
Sweden	10
Netherlands	10
Norway	9
Italy	9
Austria	9
Belgium	7

Que3. What are top 3 values of total invoice?

```sql
-- What are top 3 values of total invoice
SELECT round(total) FROM invoice
ORDER BY total desc
LIMIT 3;
```

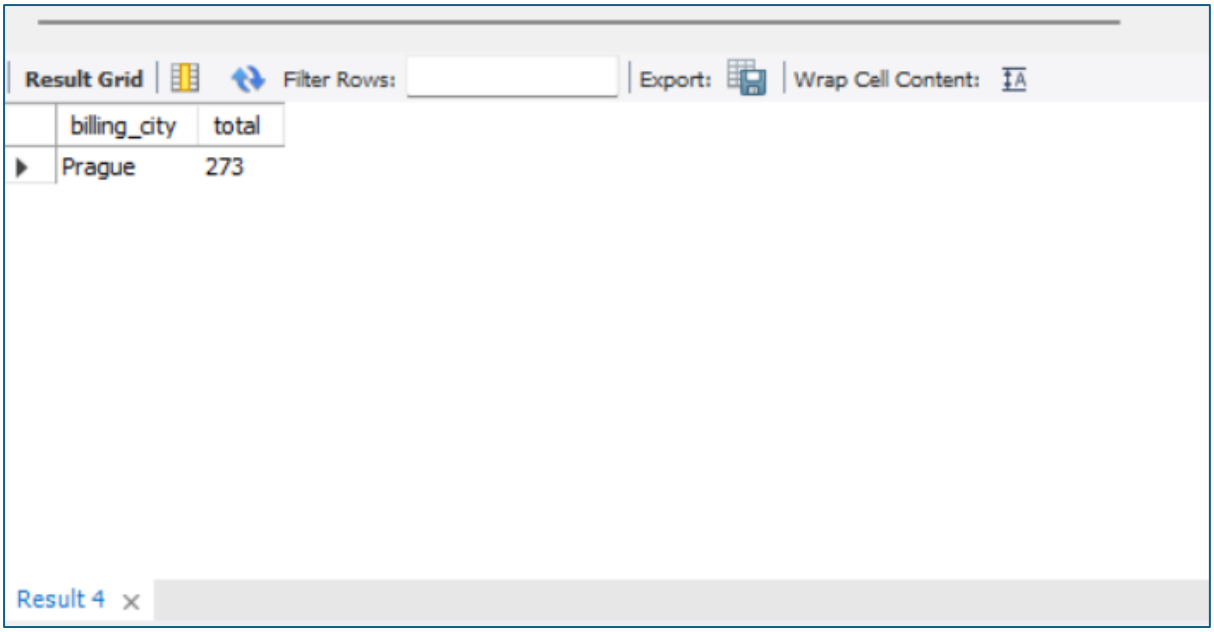
Output:



Que4. 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.

```
16    -- 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
17    -- query that returns one city that has the highest sum of invoice totals. Return both the city name and sum of invoice total.
18 •  SELECT billing_city,ROUND(sum(total))as total FROM invoice
19    GROUP BY billing_city
20    ORDER BY total desc
21    LIMIT 1;
```

Output:



Que5. 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.

```
23    -- Who is the best customers? The customer who sepnd the most money will be declared as the best customer. Write a query that returns
24    -- the person who has spend the most money.
25    SELECT concat(first_name,"",last_name) AS Name, sum(total) AS Total FROM customer c
26    INNER JOIN invoice i
27    ON i.customer_id = c.customer_id
28    GROUP BY Name
29    ORDER BY Total desc
30    LIMIT 1;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	Total
FrantiÅiekWichterlovÃi	144.54000000000002

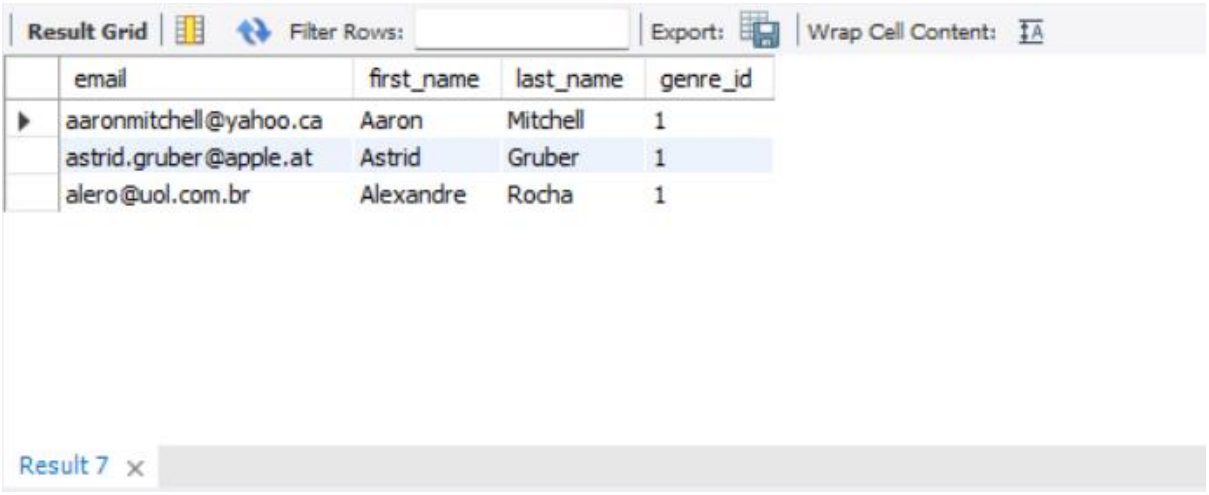
Result 5

Question Set 2 (Medium)

Que1. 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

```
32    -- Write a query to return the email, first_name,last_name, and genere of all rock music listners. Return your list ordered alphbetically
33    -- by email starting with A
34    SELECT c.email,c.first_name,c.last_name,g.genre_id FROM customer c
35    INNER JOIN invoice i
36    ON i.customer_id = c.customer_id
37    INNER JOIN invoice_line il
38    ON il.invoice_id = i.invoice_id
39    INNER JOIN track t
40    ON t.track_id = il.track_id
41    INNER JOIN genre g
42    ON g.genre_id = t.genre_id
43    WHERE g.name = 'Rock' AND
44    email LIKE "a%"
45    GROUP BY c.email,c.first_name,c.last_name,g.genre_id;
46
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

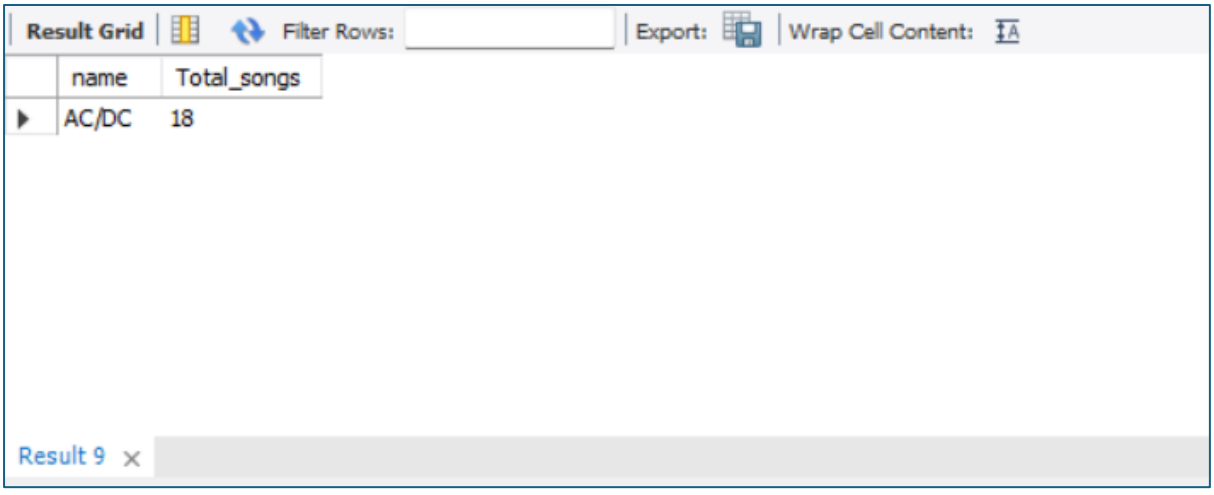
email	first_name	last_name	genre_id
aaronmitchell@yahoo.ca	Aaron	Mitchell	1
astrid.gruber@apple.at	Astrid	Gruber	1
alero@uol.com.br	Alexandre	Rocha	1

Result 7

Que2. 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.

```
47    -- Lets invite the artist who have written the most rock music in our dataset. Write a query that returns the Artist name and total
48    -- track count of the top 10 rock bands.
49 •  SELECT * FROM track;
50 •  SELECT a.name,count(t.track_id)AS Total_songs FROM artist a
51    INNER JOIN album al
52    ON al.artist_id = a.artist_id
53    INNER JOIN track t
54    ON t.album_id = al.album_id
55    INNER JOIN genre g
56    ON g.genre_id = t.genre_id
57    WHERE g.name = 'Rock'
58    GROUP BY a.name
59    ORDER BY Total_songs DESC
60    LIMIT 1;
61
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

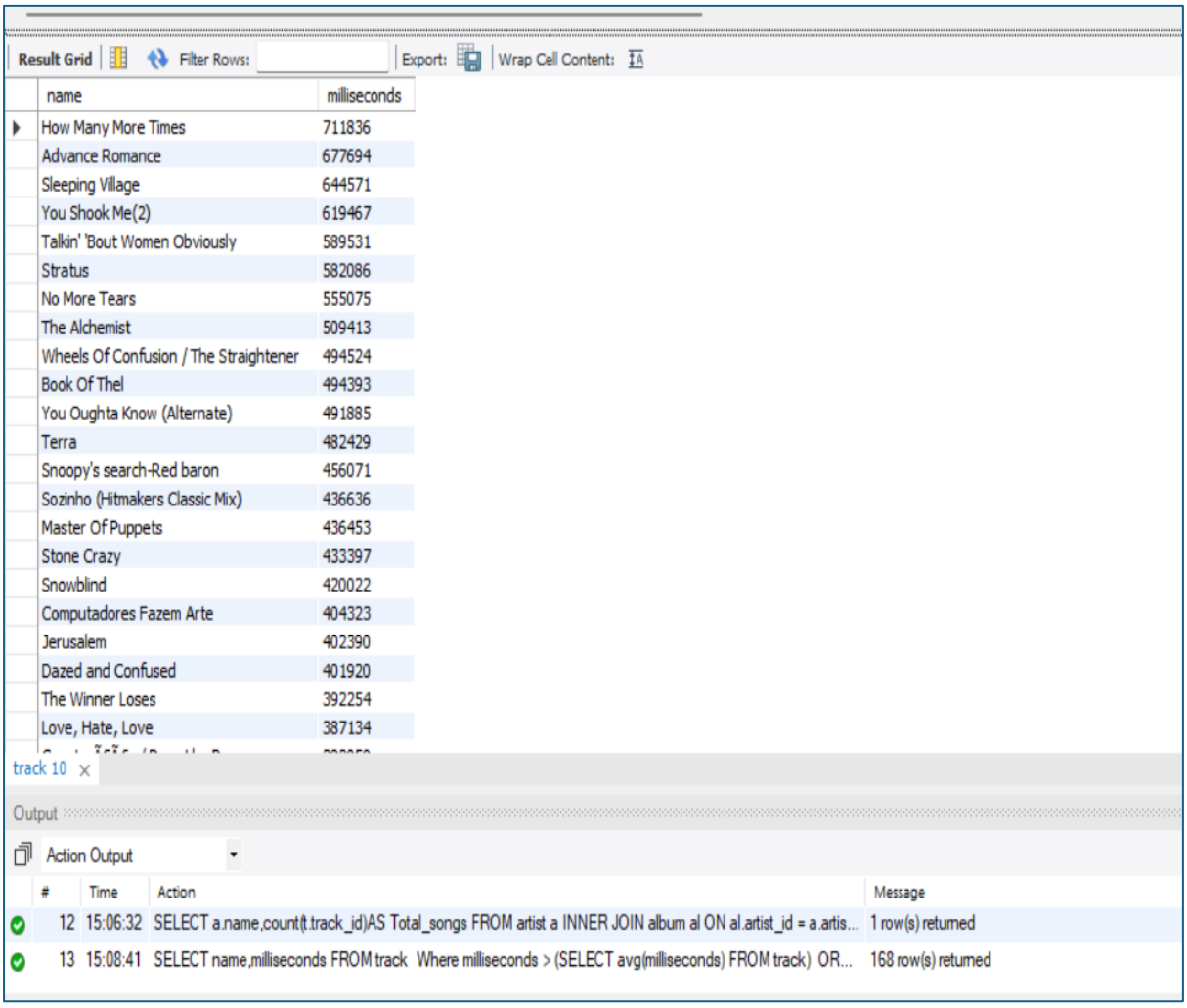
name	Total_songs
AC/DC	18

Result 9

Que3. 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

```
62    -- Return the song name that have a song length longer than the average song length. Return the name and milliseconds for each track.
63    -- Order by the song lenght with the longest song listed first.
64 •  SELECT name,milliseconds FROM track
65    Where milliseconds > (SELECT avg(milliseconds) FROM track)
66    ORDER BY milliseconds desc;
67
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	milliseconds
How Many More Times	711836
Advance Romance	677694
Sleeping Village	644571
You Shook Me(2)	619467
Talkin' 'Bout Women Obviously	589531
Stratus	582086
No More Tears	555075
The Alchemist	509413
Wheels Of Confusion / The Straightener	494524
Book Of Thel	494393
You Oughta Know (Alternate)	491885
Terra	482429
Snoopy's search-Red baron	456071
Sozinho (Hitmakers Classic Mix)	436636
Master Of Puppets	436453
Stone Crazy	433397
Snowblind	420022
Computadores Fazem Arte	404323
Jerusalem	402390
Dazed and Confused	401920
The Winner Loses	392254
Love, Hate, Love	387134

track 10 ×

Output

Action Output

#	Time	Action	Message
12	15:06:32	SELECT a.name,count(t.track_id)AS Total_songs FROM artist a INNER JOIN album al ON al.artist_id = a.artis...	1 row(s) returned
13	15:08:41	SELECT name,milliseconds FROM track Where milliseconds > (SELECT avg(milliseconds) FROM track) OR...	168 row(s) returned

Question Set 3 (Advance)

Que1. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

```sql
-- 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,artist.name,sum(invoice_line.unit_price*invoice_line.quantity)AS TOTAL FROM artist
INNER JOIN album ON album.artist_id = artist.artist_id
INNER JOIN track ON album.album_id = track.album_id
INNER JOIN invoice_line ON invoice_line.track_id = track.track_id
GROUP BY artist.name,artist.artist_id
ORDER BY TOTAL DESC
LIMIT 1
)
SELECT customer.first_name,customer.last_name,best_selling_artist.name,sum(invoice_line.unit_price*invoice_line.quantity)AS Amount FROM invoi
INNER JOIN customer ON customer.customer_id = invoice.invoice_id
INNER JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
INNER JOIN track ON track.track_id = invoice_line.track_id
INNER JOIN album ON album.album_id = track.album_id
INNER JOIN best_selling_artist ON best_selling_artist.artist_id = album.artist_id
GROUP BY customer.first_name,customer.last_name,best_selling_artist.name
ORDER BY Amount DESC;
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

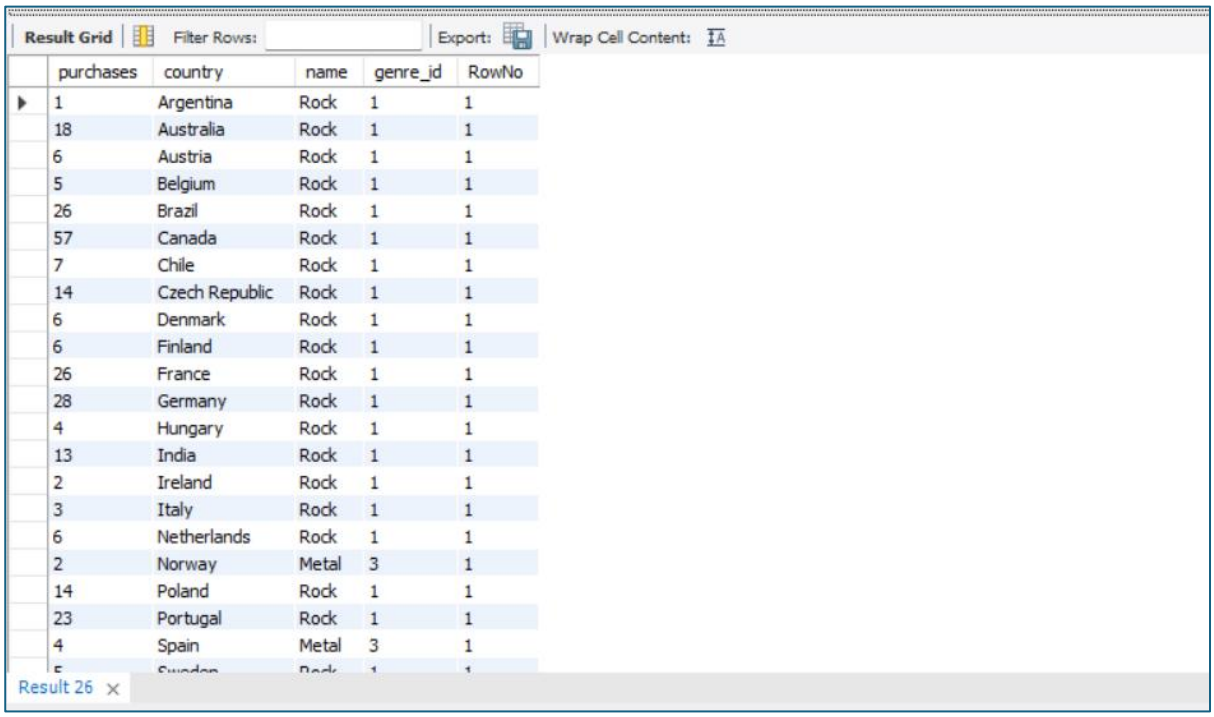
first_name	last_name	name	Amount
John	Gordon	AC/DC	9.9
Frank	Harris	AC/DC	1.98
Kathy	Chase	AC/DC	0.99
Madalena	Sampaio	AC/DC	0.99
Wyatt	Girard	AC/DC	0.99
Lucas	Mancini	AC/DC	0.99

Result 12

Que2. 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.

```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 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;
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

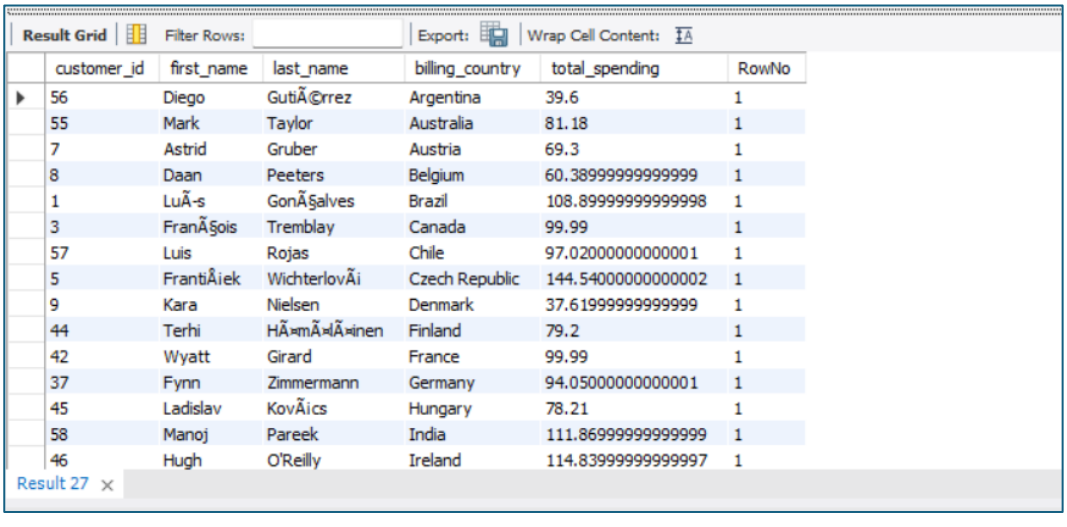
purchases	country	name	genre_id	RowNo
1	Argentina	Rock	1	1
18	Australia	Rock	1	1
6	Austria	Rock	1	1
5	Belgium	Rock	1	1
26	Brazil	Rock	1	1
57	Canada	Rock	1	1
7	Chile	Rock	1	1
14	Czech Republic	Rock	1	1
6	Denmark	Rock	1	1
6	Finland	Rock	1	1
26	France	Rock	1	1
28	Germany	Rock	1	1
4	Hungary	Rock	1	1
13	India	Rock	1	1
2	Ireland	Rock	1	1
3	Italy	Rock	1	1
6	Netherlands	Rock	1	1
2	Norway	Metal	3	1
14	Poland	Rock	1	1
23	Portugal	Rock	1	1
4	Spain	Metal	3	1

Result 26

Que3. 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.

```
105    -- Write a query that determines the customer that has spent the most on music for each
106    -- country. Write a query that returns the country along with the top customer and how
107    -- much they spent. For countries where the top amount spent is shared, provide all
108    -- customers who spent this amount
109    WITH Customter_with_country AS (
110            SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
111            ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
112            FROM invoice
113            JOIN customer ON customer.customer_id = invoice.customer_id
114            GROUP BY 1,2,3,4
115            ORDER BY 4 ASC,5 DESC)
116    SELECT * FROM Customter_with_country WHERE RowNo <= 1;
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	billing_country	total_spending	RowNo
56	Diego	GutiÃ©rrez	Argentina	39.6	1
55	Mark	Taylor	Australia	81.18	1
7	Astrid	Gruber	Austria	69.3	1
8	Daan	Peeters	Belgium	60.38999999999999	1
1	LuÃ-s	GonÃ§alves	Brazil	108.89999999999998	1
3	FranÃ§ois	Tremblay	Canada	99.99	1
57	Luis	Rojas	Chile	97.02000000000001	1
5	FrantiÅiek	WichterlovÃi	Czech Republic	144.54000000000002	1
9	Kara	Nielsen	Denmark	37.61999999999999	1
44	Terhi	HÃ¤mÃ¤lÃ¤inen	Finland	79.2	1
42	Wyatt	Girard	France	99.99	1
37	Fynn	Zimmermann	Germany	94.05000000000001	1
45	Ladislav	KovÃics	Hungary	78.21	1
58	Manoj	Pareek	India	111.86999999999999	1
46	Hugh	O'Reilly	Ireland	114.83999999999997	1

Result 27