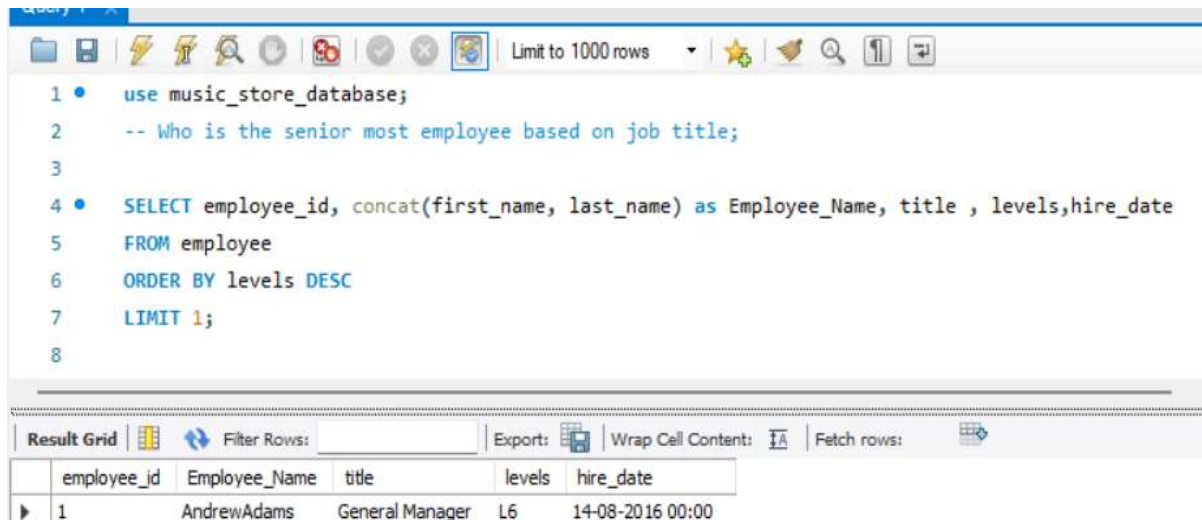


Project : MySQL Music Store Database

SECTION : EASY

QUERY 1 :



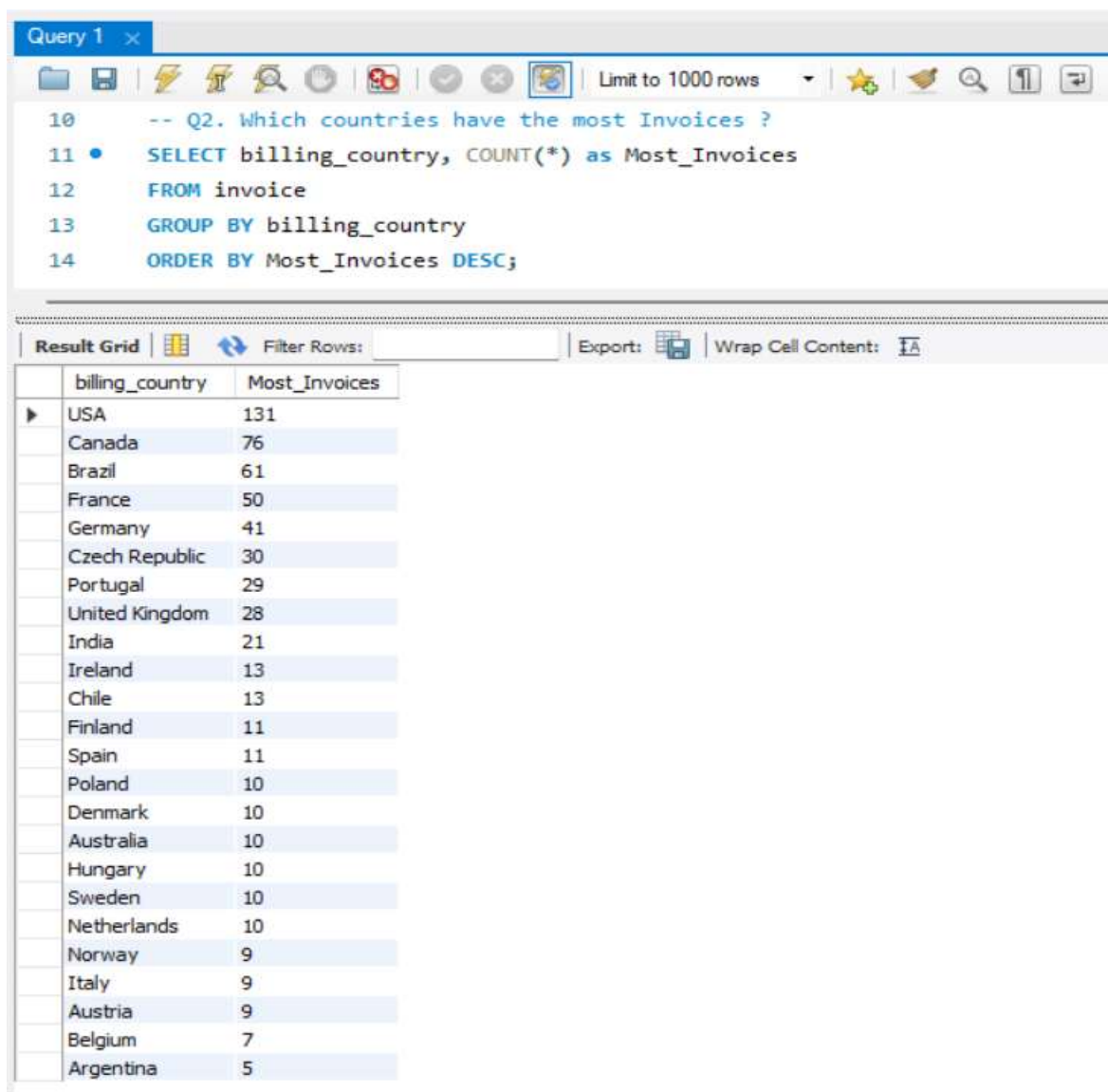
The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, execution, and navigation. The SQL editor contains the following query:

```
1 • use music_store_database;
2 -- Who is the senior most employee based on job title;
3
4 • SELECT employee_id, concat(first_name, last_name) as Employee_Name, title , levels,hire_date
5 FROM employee
6 ORDER BY levels DESC
7 LIMIT 1;
8
```

Below the editor, the 'Result Grid' tab is active, displaying the results of the query in a table format:

	employee_id	Employee_Name	title	levels	hire_date
▶	1	AndrewAdams	General Manager	L6	14-08-2016 00:00

QUERY 2 :



The screenshot shows the MySQL Workbench interface. The top toolbar is identical to the previous screenshot. The SQL editor contains the following query:

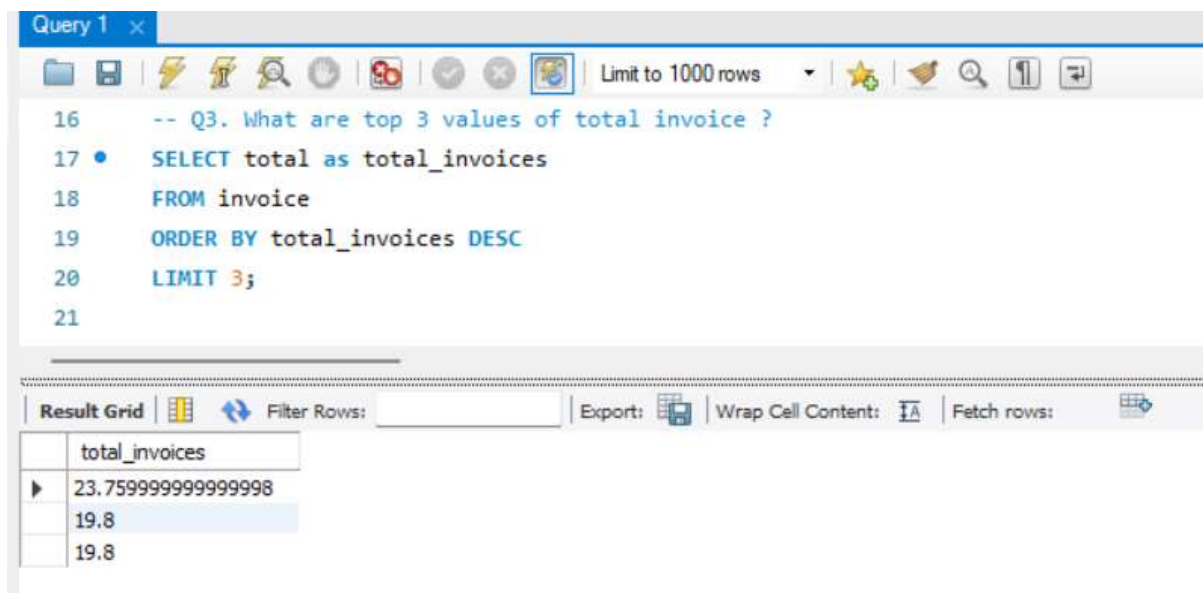
```
10 -- Q2. Which countries have the most Invoices ?
11 • SELECT billing_country, COUNT(*) as Most_Invoices
12 FROM invoice
13 GROUP BY billing_country
14 ORDER BY Most_Invoices DESC;
```

Below the editor, the 'Result Grid' tab is active, displaying the results of the query in a table format:

	billing_country	Most_Invoices
▶	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
	Argentina	5

Project : MySQL Music Store Database

QUERY 3 :



Query 1 x

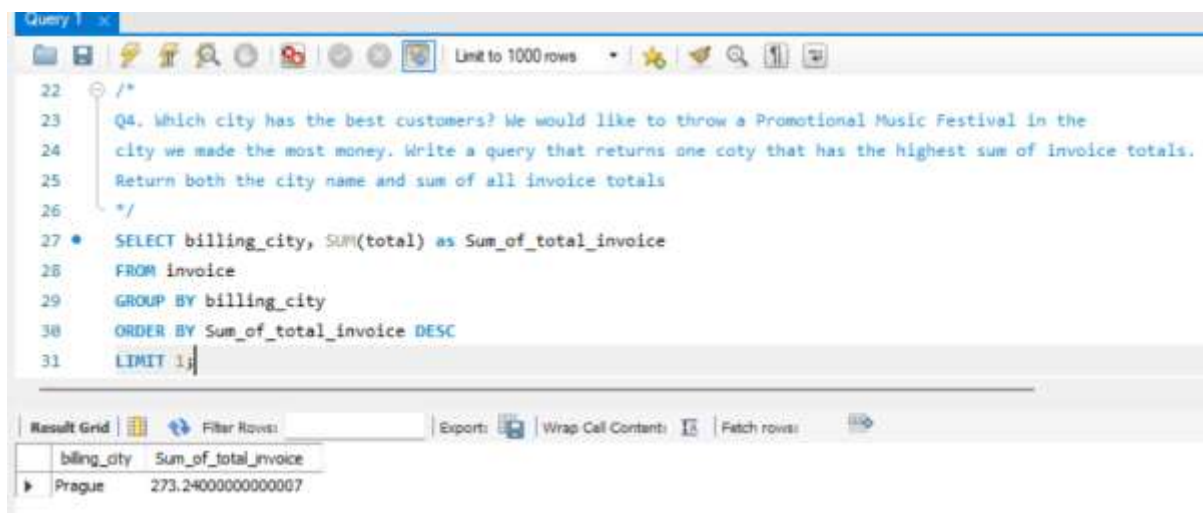
Limit to 1000 rows

```
16 -- Q3. What are top 3 values of total invoice ?
17 • SELECT total as total_invoices
18 FROM invoice
19 ORDER BY total_invoices DESC
20 LIMIT 3;
21
```

Result Grid

	total_invoices
▶	23.759999999999998
	19.8
	19.8

QUERY 4 :



Query 1 x

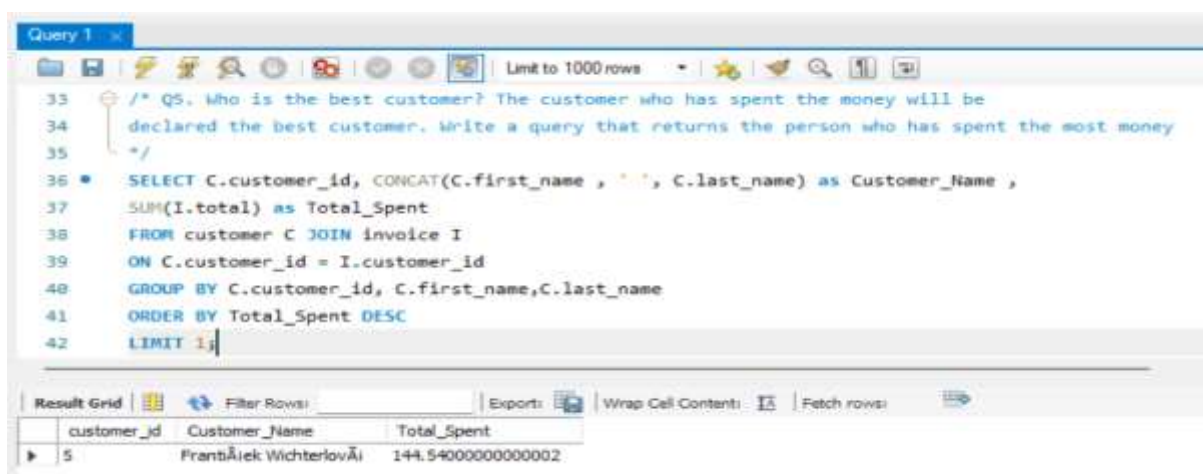
Limit to 1000 rows

```
22 /*
23 Q4. Which city has the best customers? We would like to throw a Promotional Music Festival in the
24 city we made the most money. Write a query that returns one city that has the highest sum of invoice totals.
25 Return both the city name and sum of all invoice totals
26 */
27 • SELECT billing_city, SUM(total) as Sum_of_total_invoice
28 FROM invoice
29 GROUP BY billing_city
30 ORDER BY Sum_of_total_invoice DESC
31 LIMIT 1;
```

Result Grid

	billing_city	Sum_of_total_invoice
▶	Prague	273.24000000000007

QUERY 5 :



Query 1 x

Limit to 1000 rows

```
33 /* Q5. Who is the best customer? The customer who has spent the money will be
34 declared the best customer. Write a query that returns the person who has spent the most money
35 */
36 • SELECT C.customer_id, CONCAT(C.first_name , ' ', C.last_name) as Customer_Name ,
37 SUM(I.total) as Total_Spent
38 FROM customer C JOIN invoice I
39 ON C.customer_id = I.customer_id
40 GROUP BY C.customer_id, C.first_name, C.last_name
41 ORDER BY Total_Spent DESC
42 LIMIT 1;
```

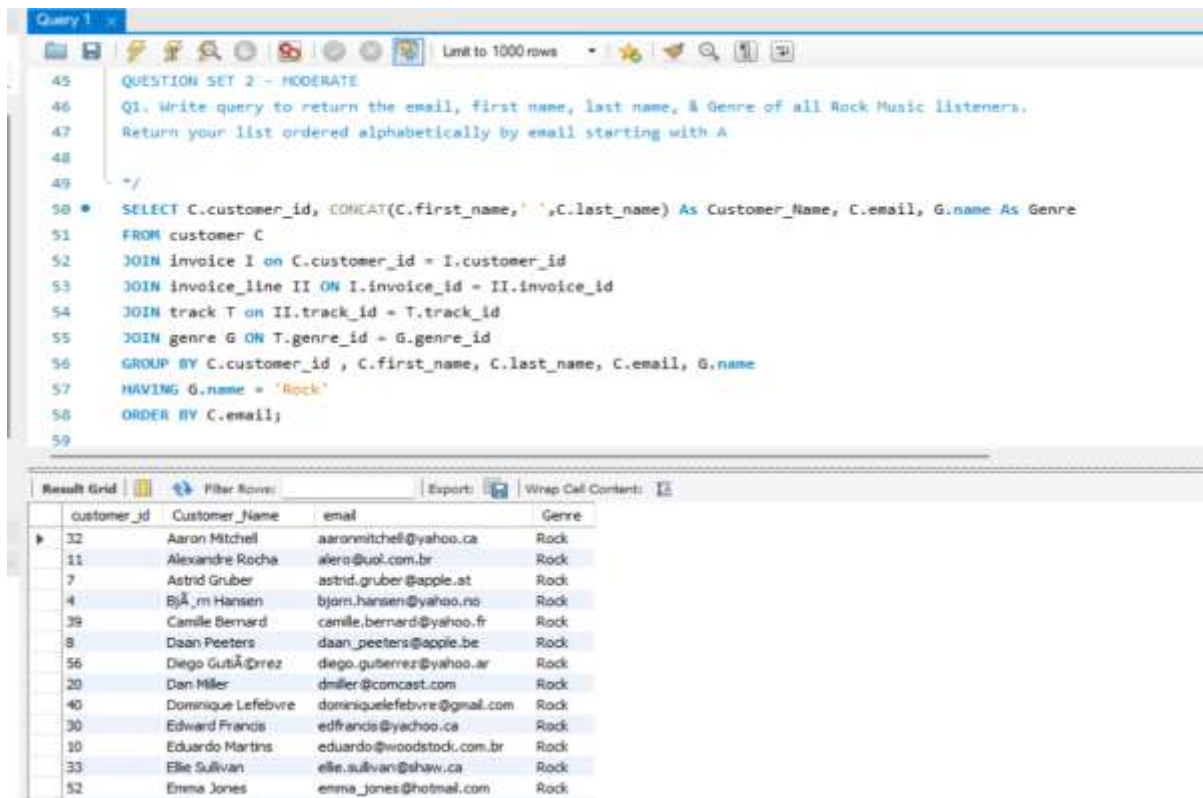
Result Grid

	customer_id	Customer_Name	Total_Spent
▶	5	František WichterlovÁi	144.54000000000002

Project : MySQL Music Store Database

SECTION : MODERATE

QUERY 1 :



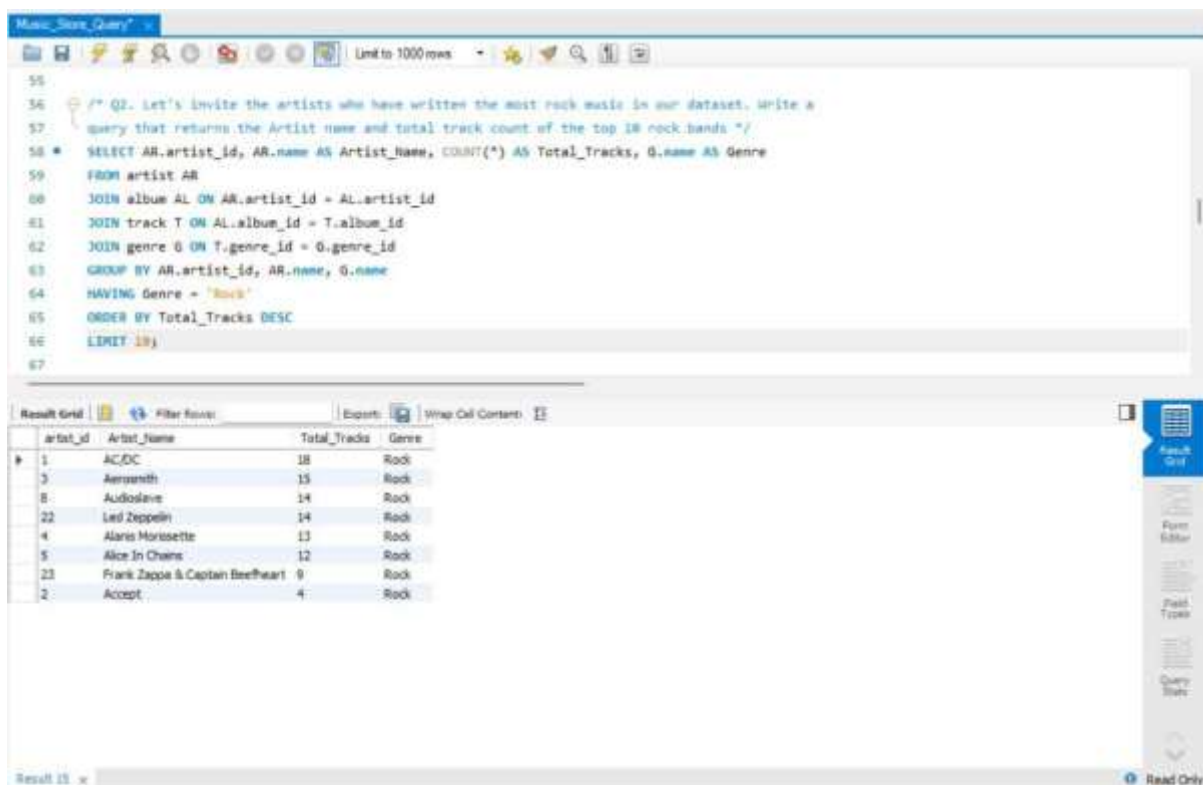
Query 1 - MODERATE

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

```
*/
SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) As Customer_Name, C.email, G.name As Genre
FROM customer C
JOIN invoice I ON C.customer_id = I.customer_id
JOIN invoice_line II ON I.invoice_id = II.invoice_id
JOIN track T ON II.track_id = T.track_id
JOIN genre G ON T.genre_id = G.genre_id
GROUP BY C.customer_id, C.first_name, C.last_name, C.email, G.name
HAVING G.name = 'Rock'
ORDER BY C.email;
```

customer_id	Customer_Name	email	Genre
32	Aaron Mitchell	aaronmitchell@yahoo.ca	Rock
11	Alexandre Rocha	alex@uol.com.br	Rock
7	Astrid Gruber	astrid.gruber@apple.at	Rock
4	Björn Hansen	bjorn.hansen@yahoo.no	Rock
39	Camille Bernard	camille.bernard@yahoo.fr	Rock
8	Daan Peeters	daan.peeters@apple.be	Rock
56	Diego Gutierrez	diego.gutierrez@yahoo.ar	Rock
20	Dan Miller	dmiller@comcast.com	Rock
40	Dominique Lefebvre	dominiquelefebvre@gmail.com	Rock
30	Edward Francis	edfrancis@yahoo.ca	Rock
10	Eduardo Martins	eduardo@woodstock.com.br	Rock
33	Elle Sullivan	elle.sullivan@shaw.ca	Rock
52	Emma Jones	emma_jones@hotmail.com	Rock

QUERY 2 :



Music_Store_Query

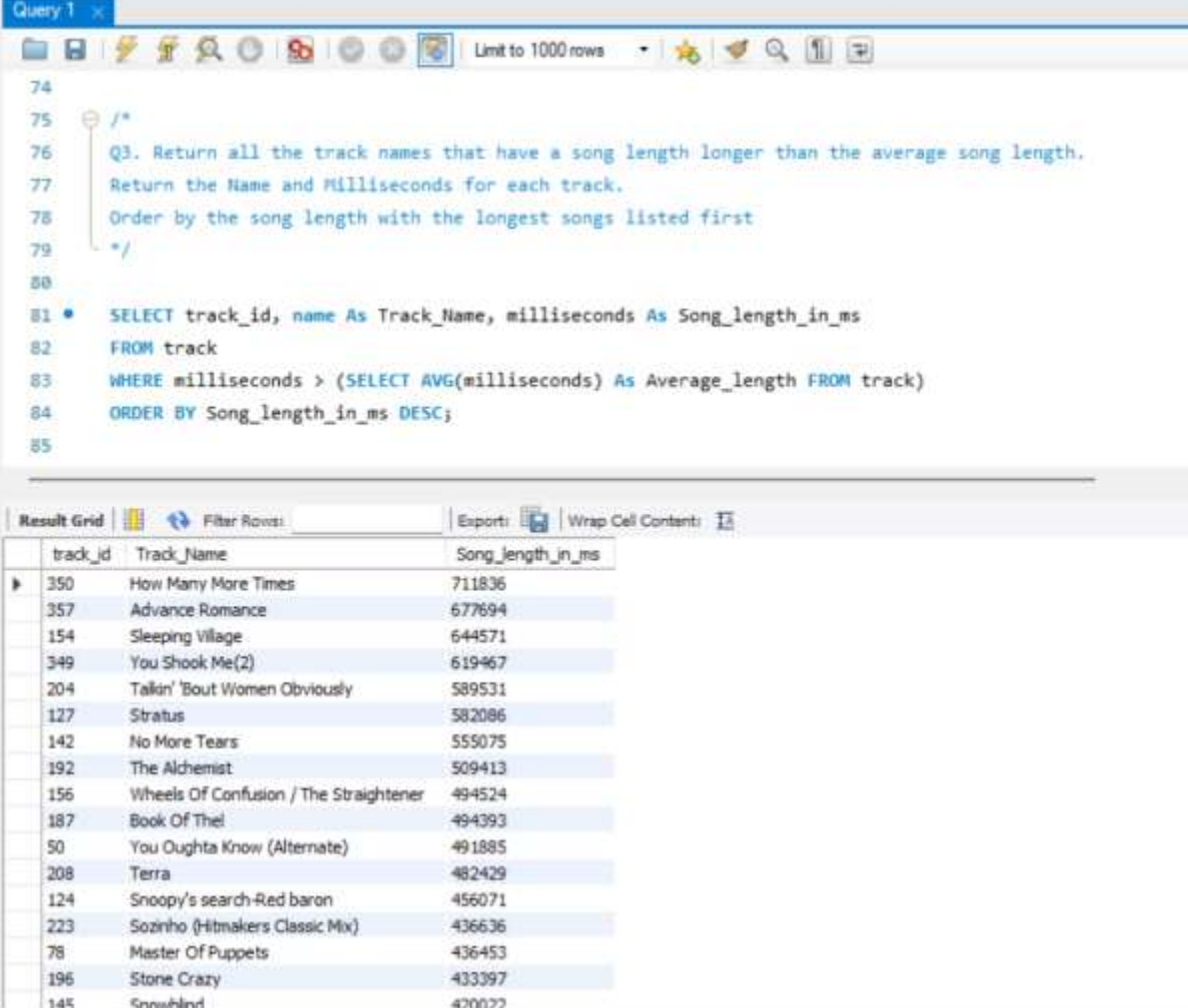
Q2. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands

```
*/
SELECT AR.artist_id, AR.name AS Artist_Name, COUNT(*) AS Total_Tracks, G.name AS Genre
FROM artist AR
JOIN album AL ON AR.artist_id = AL.artist_id
JOIN track T ON AL.album_id = T.album_id
JOIN genre G ON T.genre_id = G.genre_id
GROUP BY AR.artist_id, AR.name, G.name
HAVING Genre = 'Rock'
ORDER BY Total_Tracks DESC
LIMIT 10;
```

artist_id	Artist_Name	Total_Tracks	Genre
1	AC/DC	18	Rock
3	Aerosmith	15	Rock
8	Audioslave	14	Rock
22	Led Zeppelin	14	Rock
4	Alanis Morissette	13	Rock
5	Alice In Chains	12	Rock
23	Frank Zappa & Captain Beefheart	9	Rock
2	Accept	4	Rock

Project : MySQL Music Store Database

QUERY 3 :



The screenshot shows a MySQL query editor window titled "Query 1". The query is as follows:

```
74
75 /*
76 Q3. Return all the track names that have a song length longer than the average song length.
77 Return the Name and Milliseconds for each track.
78 Order by the song length with the longest songs listed first
79 */
80
81 • SELECT track_id, name As Track_Name, milliseconds As Song_length_in_ms
82 FROM track
83 WHERE milliseconds > (SELECT AVG(milliseconds) As Average_length FROM track)
84 ORDER BY Song_length_in_ms DESC;
85
```

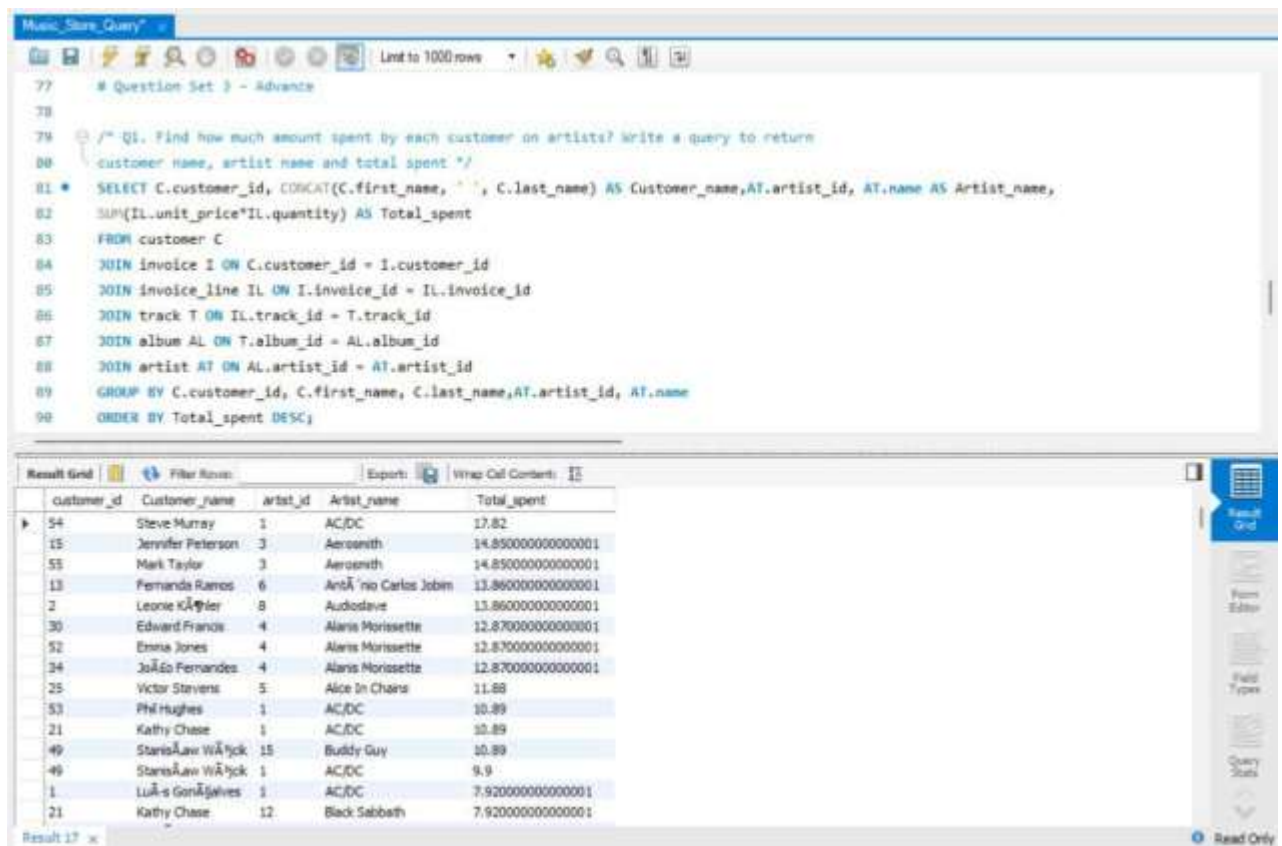
Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has four columns: track_id, Track_Name, and Song_length_in_ms. The results are sorted by Song_length_in_ms in descending order.

track_id	Track_Name	Song_length_in_ms
350	How Many More Times	711836
357	Advance Romance	677694
154	Sleeping Village	644571
349	You Shook Me(2)	619467
204	Takin' 'Bout Women Obviously	589531
127	Stratus	582086
142	No More Tears	555075
192	The Alchemist	509413
156	Wheels Of Confusion / The Straightener	494524
187	Book Of Thel	494393
50	You Oughta Know (Alternate)	491885
208	Terra	482429
124	Snoopy's search-Red baron	456071
223	Sozinho (Hitmakers Classic Mix)	436636
78	Master Of Puppets	436453
196	Stone Crazy	433397
145	Snowblind	420072

Project : MySQL Music Store Database

SECTION 3 : ADVANCE

QUERY 1 :



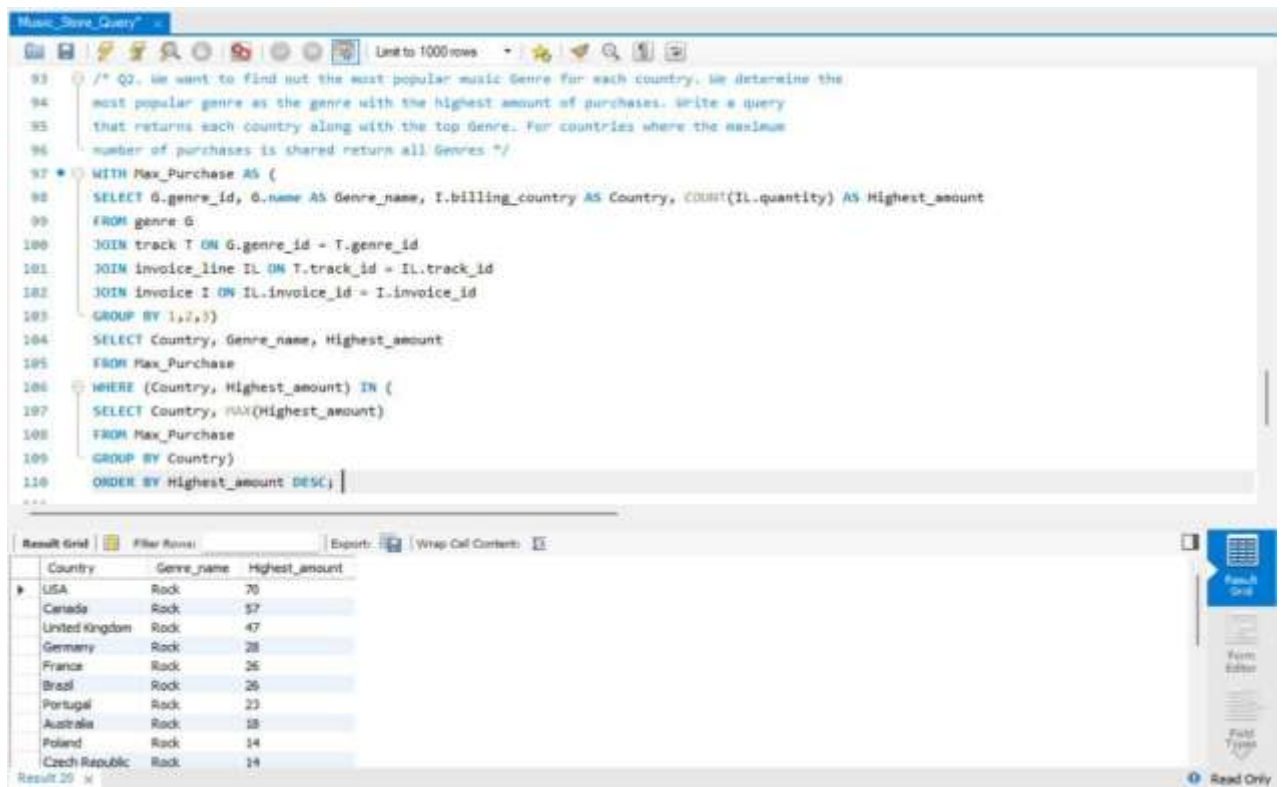
The screenshot shows the MySQL Music Store Query Editor interface. The query editor displays a SQL query for Query 1, which aims to find the total amount spent by each customer on artists. The query is as follows:

```
/* Q1. Find how much amount spent by each customer on artists? Write a query to return
customer name, artist name and total spent */
SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, AT.artist_id, AT.name AS Artist_name,
SUM(IL.unit_price*IL.quantity) AS Total_spent
FROM customer C
JOIN invoice I ON C.customer_id = I.customer_id
JOIN invoice_line IL ON I.invoice_id = IL.invoice_id
JOIN track T ON IL.track_id = T.track_id
JOIN album AL ON T.album_id = AL.album_id
JOIN artist AT ON AL.artist_id = AT.artist_id
GROUP BY C.customer_id, C.first_name, C.last_name, AT.artist_id, AT.name
ORDER BY Total_spent DESC;
```

The results grid shows the following data:

customer_id	Customer_name	artist_id	Artist_name	Total_spent
54	Steve Murray	1	AC/DC	17.82
15	Jennifer Peterson	3	Aerosmith	14.850000000000001
55	Mark Taylor	3	Aerosmith	14.850000000000001
13	Fernanda Ramos	6	Ana'nio Carlos Jobim	13.860000000000001
2	Leonie Kiefer	8	Audioslave	13.860000000000001
30	Edward Francis	4	Alain Morissette	12.870000000000001
52	Enna Jones	4	Alain Morissette	12.870000000000001
24	João Fernandes	4	Alain Morissette	12.870000000000001
25	Victor Stevens	5	Alice In Chains	11.88
53	Phil Hughes	1	AC/DC	10.89
21	Kathy Chase	1	AC/DC	10.89
49	Stars & W&A	15	Buddy Guy	10.89
49	Stars & W&A	1	AC/DC	9.9
1	Luis Goncalves	1	AC/DC	7.920000000000001
21	Kathy Chase	12	Black Sabbath	7.920000000000001

QUERY 2 :



The screenshot shows the MySQL Music Store Query Editor interface. The query editor displays a SQL query for Query 2, which aims to find the most popular music genre for each country based on the highest amount of purchases. The query is as follows:

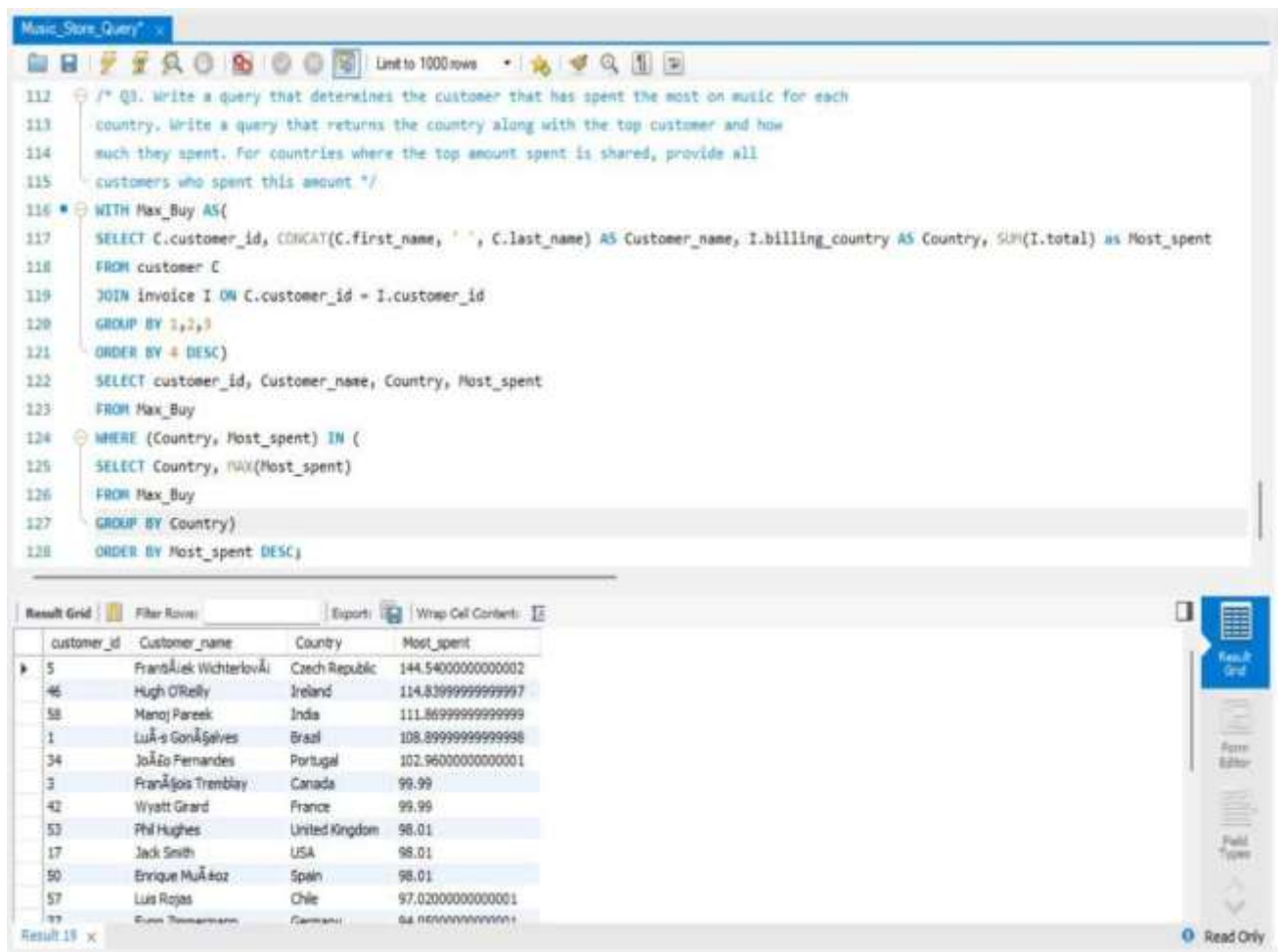
```
/* Q2. We want to find out the most popular music Genre for each country. We determine the
most popular genre as the genre with the highest amount of purchases. Write a query
that returns each country along with the top Genre. For countries where the maximum
number of purchases is shared return all Genres */
WITH Max_Purchase AS (
SELECT G.genre_id, G.name AS Genre_name, I.billing_country AS Country, COUNT(IL.quantity) AS Highest_amount
FROM genre G
JOIN track T ON G.genre_id = T.genre_id
JOIN invoice_line IL ON T.track_id = IL.track_id
JOIN invoice I ON IL.invoice_id = I.invoice_id
GROUP BY 1,2,3)
SELECT Country, Genre_name, Highest_amount
FROM Max_Purchase
WHERE (Country, Highest_amount) IN (
SELECT Country, MAX(Highest_amount)
FROM Max_Purchase
GROUP BY Country)
ORDER BY Highest_amount DESC;
```

The results grid shows the following data:

Country	Genre_name	Highest_amount
USA	Rock	70
Canada	Rock	57
United Kingdom	Rock	47
Germany	Rock	28
France	Rock	26
Brazil	Rock	26
Portugal	Rock	23
Australia	Rock	18
Poland	Rock	14
Czech Republic	Rock	14

Project : MySQL Music Store Database

QUERY 3 :



The screenshot displays a MySQL IDE window titled "Music_Store_Query". The SQL editor contains a query designed to find the top customer by spending amount for each country. The query uses a CTE named "Max_Buy" to first identify the top customer per country, and then a final SELECT to list these customers. The results are shown in a table with columns: customer_id, Customer_name, Country, and Most_spent.

```
112 /* Q3. Write a query that determines the customer that has spent the most on music for each
113 country. Write a query that returns the country along with the top customer and how
114 much they spent. For countries where the top amount spent is shared, provide all
115 customers who spent this amount */
116 WITH Max_Buy AS(
117     SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS Customer_name, I.billing_country AS Country, SUM(I.total) as Most_spent
118     FROM customer C
119     JOIN invoice I ON C.customer_id = I.customer_id
120     GROUP BY 1,2,3
121     ORDER BY 4 DESC)
122 SELECT customer_id, Customer_name, Country, Most_spent
123 FROM Max_Buy
124 WHERE (Country, Most_spent) IN (
125     SELECT Country, MAX(Most_spent)
126     FROM Max_Buy
127     GROUP BY Country)
128 ORDER BY Most_spent DESC;
```

customer_id	Customer_name	Country	Most_spent
5	František Wichterlovský	Czech Republic	144.54000000000002
46	Hugh O'Reilly	Ireland	114.83999999999997
58	Manoj Pareek	India	111.86999999999999
1	Luís Gonçalves	Brazil	108.89999999999998
34	João Fernandes	Portugal	102.96000000000001
3	François Tremblay	Canada	99.99
42	Wyatt Girard	France	99.99
53	Phil Hughes	United Kingdom	98.01
17	Jack Smith	USA	98.01
50	Enrique Muñoz	Spain	98.01
57	Luis Rojas	Chile	97.02000000000001
99	Frank Thomasmann	Germany	94.96000000000001