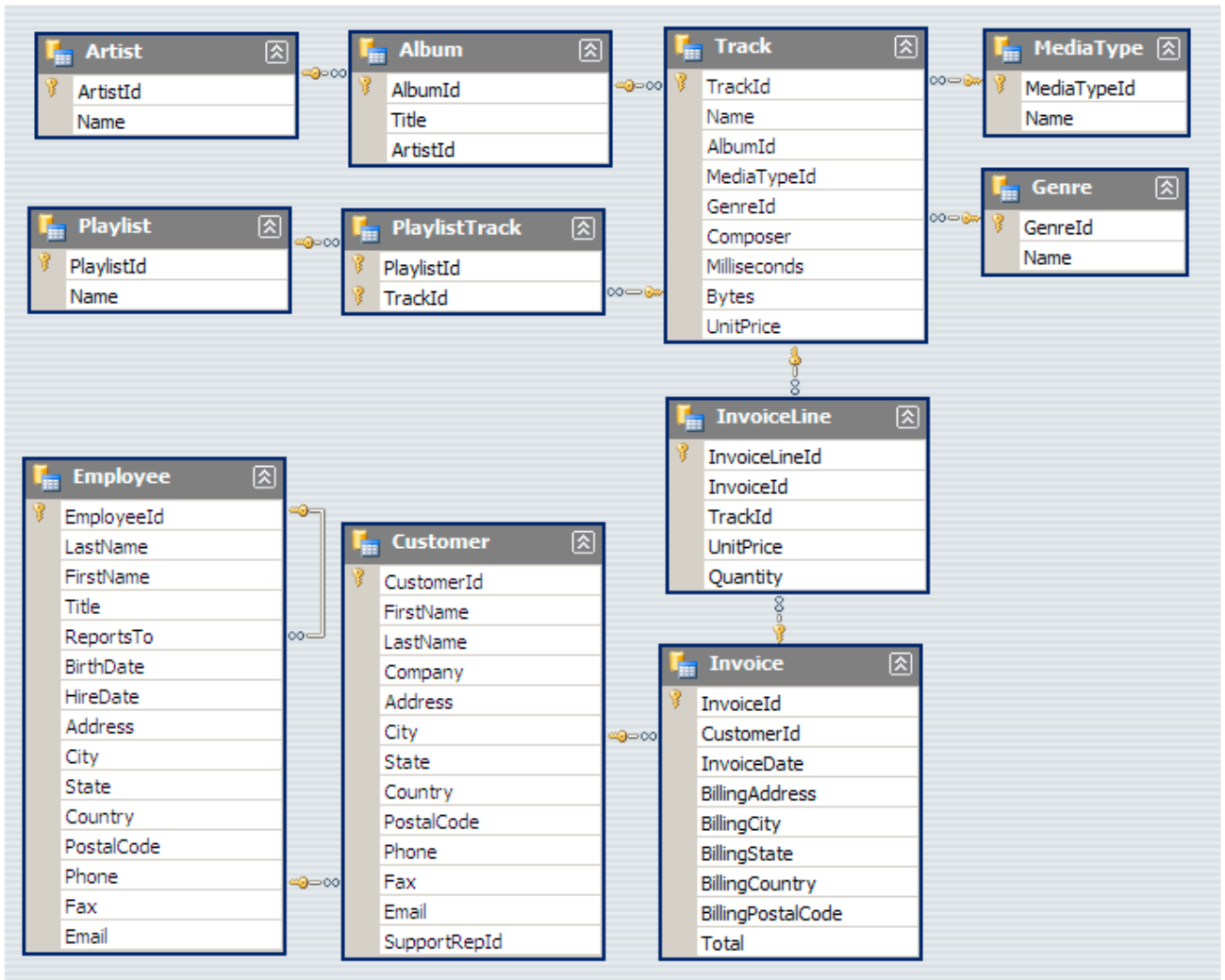


## SQL MUSIC STORE ANALYSIS

### PROJECT SCHEMA



Q1: Who is the senior most employee based on job title?



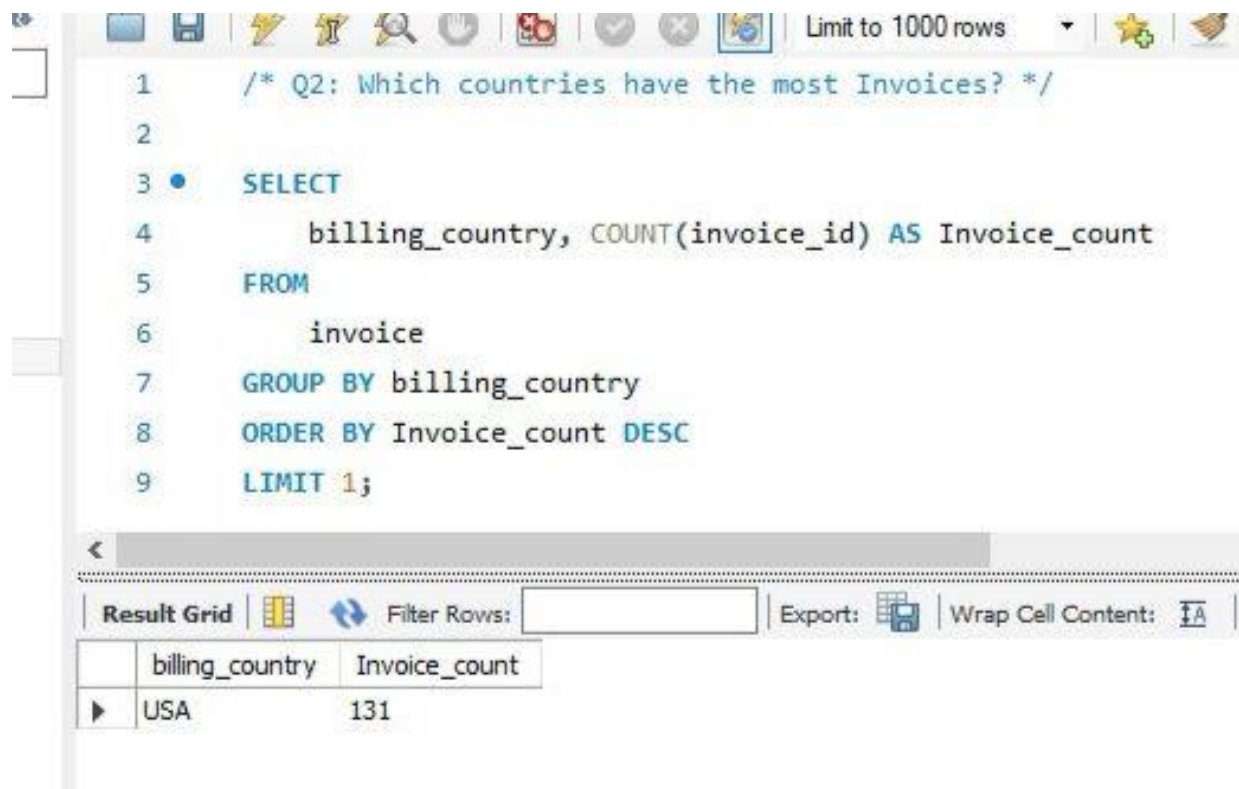
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 • SELECT * FROM musicstore.employee;
2
3 • select employee_id, first_name, last_name from employee
4   order by levels desc
5   limit 1;
```

Below the editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

	employee_id	first_name	last_name
▶	1	Andrew	Adams

Q2: Which countries have the most Invoices?



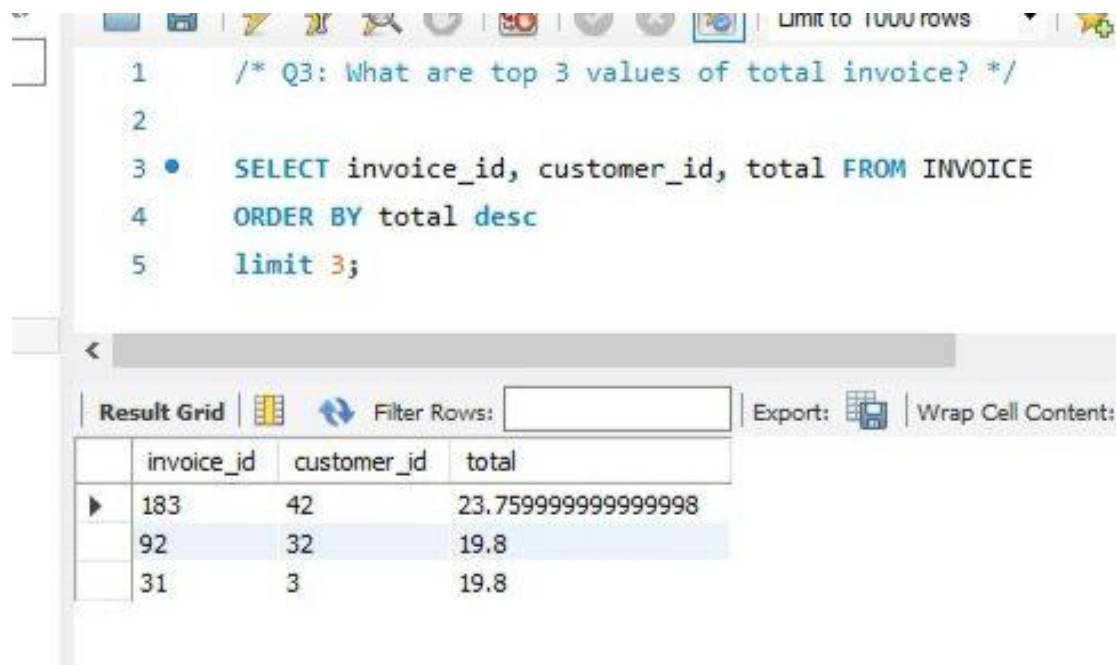
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 /* Q2: Which countries have the most Invoices? */
2
3 • SELECT
4     billing_country, COUNT(invoice_id) AS Invoice_count
5 FROM
6     invoice
7 GROUP BY billing_country
8 ORDER BY Invoice_count DESC
9 LIMIT 1;
```

Below the editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

	billing_country	Invoice_count
▶	USA	131

Q3: What are top 3 values of total invoice?



```
1  /* Q3: What are top 3 values of total invoice? */
2
3  •  SELECT invoice_id, customer_id, total FROM INVOICE
4     ORDER BY total desc
5     limit 3;
```

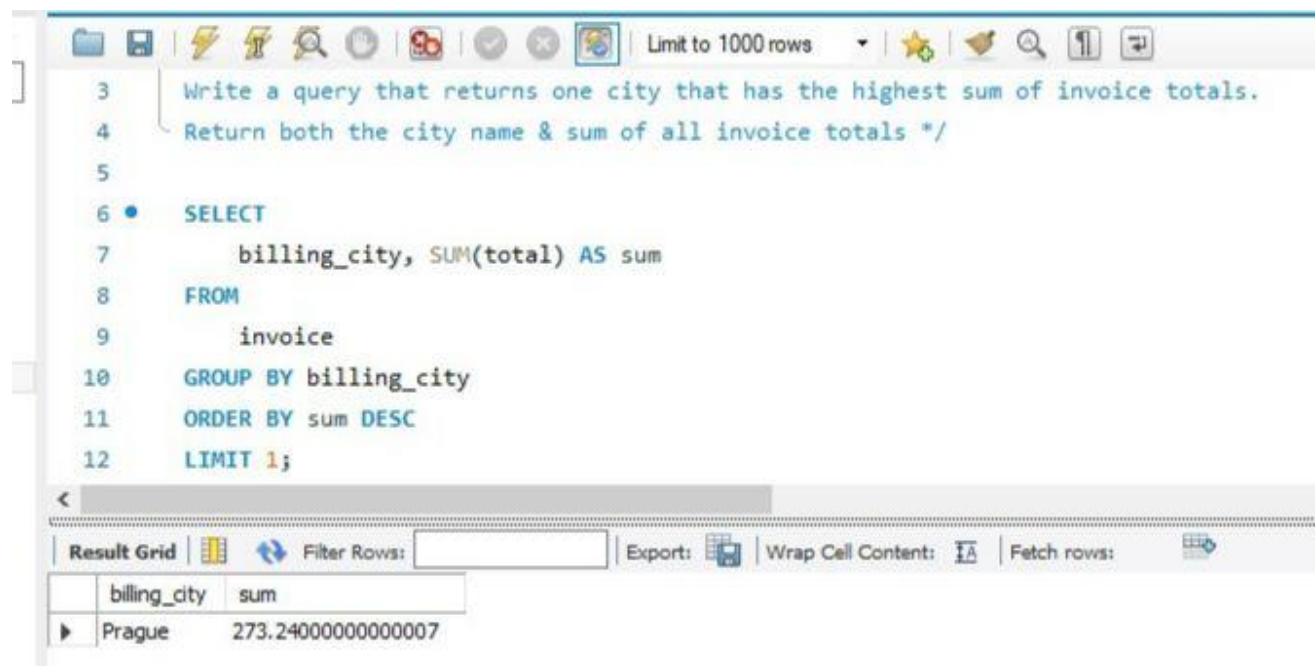
Result Grid

	invoice_id	customer_id	total
▶	183	42	23.759999999999998
	92	32	19.8
	31	3	19.8

Q4: 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



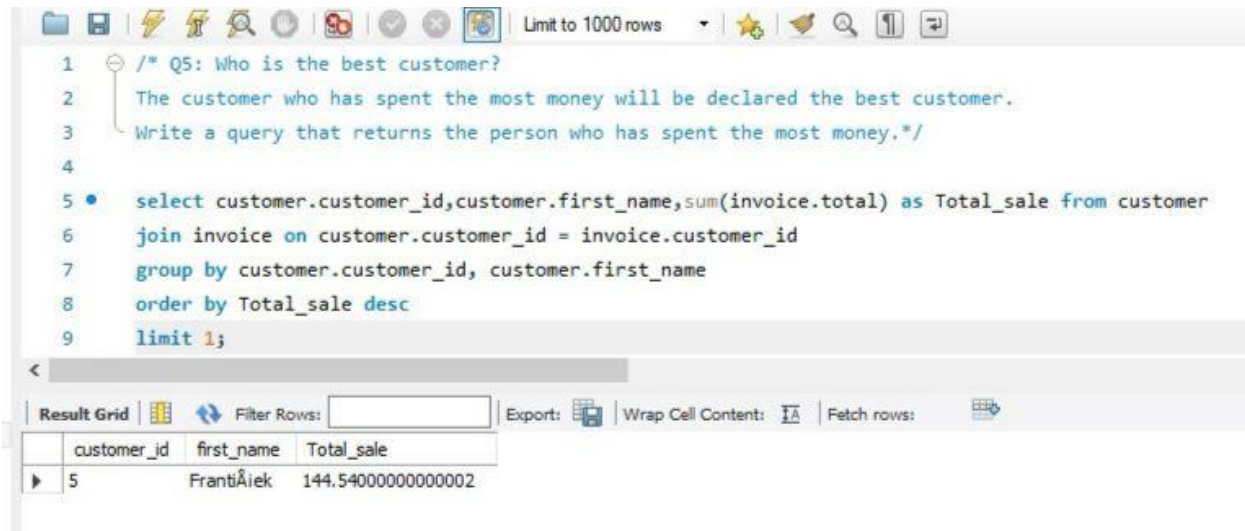
```
3  Write a query that returns one city that has the highest sum of invoice totals.
4  Return both the city name & sum of all invoice totals */
5
6  •  SELECT
7     billing_city, SUM(total) AS sum
8  FROM
9     invoice
10 GROUP BY billing_city
11 ORDER BY sum DESC
12 LIMIT 1;
```

Result Grid

	billing_city	sum
▶	Prague	273.24000000000007

**Q5: 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.**



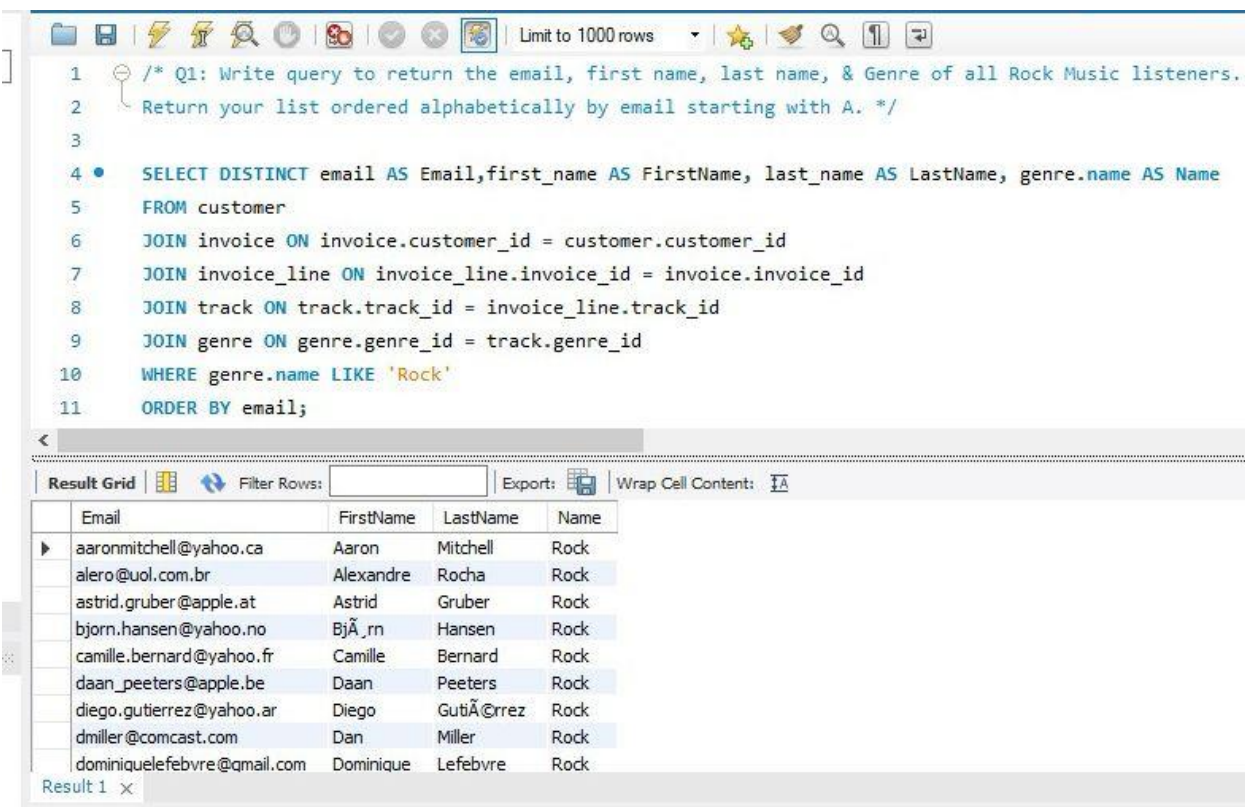
```
1  /* Q5: Who is the best customer?
2  The customer who has spent the most money will be declared the best customer.
3  Write a query that returns the person who has spent the most money.*/
4
5  • select customer.customer_id, customer.first_name, sum(invoice.total) as Total_sale from customer
6  join invoice on customer.customer_id = invoice.customer_id
7  group by customer.customer_id, customer.first_name
8  order by Total_sale desc
9  limit 1;
```

Result Grid

	customer_id	first_name	Total_sale
▶	5	František	144.54000000000002

**Q6: 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.**



```
1  /* Q1: Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
2  Return your list ordered alphabetically by email starting with A. */
3
4  • SELECT DISTINCT email AS Email, first_name AS FirstName, last_name AS LastName, genre.name AS Name
5  FROM customer
6  JOIN invoice ON invoice.customer_id = customer.customer_id
7  JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
8  JOIN track ON track.track_id = invoice_line.track_id
9  JOIN genre ON genre.genre_id = track.genre_id
10 WHERE genre.name LIKE 'Rock'
11 ORDER BY email;
```

Result Grid

	Email	FirstName	LastName	Name
▶	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
	alero@uol.com.br	Alexandre	Rocha	Rock
	astrid.gruber@apple.at	Astrid	Gruber	Rock
	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
	camille.bernard@yahoo.fr	Camille	Bernard	Rock
	daan.peeters@apple.be	Daan	Peeters	Rock
	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock
	dmiller@comcast.com	Dan	Miller	Rock
	dominiquelefebvre@gmail.com	Dominique	Lefebvre	Rock

Result 1 x

**Q7: 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.**

```
1  /* Q2: Let's invite the artists who have written the most rock music in our dataset.
2  Write a query that returns the Artist name and total track count of the top 10 rock bands. */
3
4  • SELECT artist.artist_id, artist.name, COUNT(track.track_id) AS number_of_songs
5  FROM track
6  JOIN album2 ON album2.album_id = track.album_id
7  JOIN artist ON artist.artist_id = album2.artist_id
8  JOIN genre ON genre.genre_id = track.genre_id
9  WHERE genre.name LIKE 'Rock'
10 GROUP BY artist.artist_id, artist.name
11 ORDER BY number_of_songs DESC
12 LIMIT 10;
```

Result Grid

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

**Q8: 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.**

```
5
6  • SELECT name, milliseconds
7  FROM track
8  WHERE milliseconds > (
9      SELECT AVG(milliseconds) AS avg_track_length
10     FROM track )
11 ORDER BY milliseconds DESC;
```

Result Grid

	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

track 1 ×

Output



**Q9: Find how much amount spent by each customer on Top artist ? Write a query to return customer name, artist name and total spent**

```

1
2  /* Q1: Find how much amount spent by each customer on artists?
3   Write a query to return customer name, artist name and total spend */
4
5  WITH best_selling_artist AS (
6      SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
7             SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
8      FROM invoice_line
9      JOIN track ON track.track_id = invoice_line.track_id
10     JOIN album2 ON album2.album_id = track.album_id
11     JOIN artist ON artist.artist_id = album2.artist_id
12     GROUP BY 1,2
13     ORDER BY 3 DESC
14     LIMIT 1
15 )
16 SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
17 FROM invoice i
18 JOIN customer c ON c.customer_id = i.customer_id
19 JOIN invoice_line il ON il.invoice_id = i.invoice_id
20 JOIN track t ON t.track_id = il.track_id
21 JOIN album2 alb ON alb.album_id = t.album_id
22 JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
23 GROUP BY 1,2,3,4
24 ORDER BY 5 DESC;

```

**OUTPUT:**

customer_id	first_name	last_name	artist_name	amount_spent
54	Steve	Murray	AC/DC	17.82
53	Phil	Hughes	AC/DC	10.89
21	Kathy	Chase	AC/DC	10.89
49	Stanisław	Wójcik	AC/DC	9.9
1	Luís	Gonzálves	AC/DC	7.920000000000001
24	Frank	Ralston	AC/DC	7.920000000000001
31	Martha	Silk	AC/DC	3.96
16	Frank	Harris	AC/DC	2.9699999999999998
42	Wyatt	Girard	AC/DC	2.9699999999999998
6	Helena	Holm	AC/DC	2.9699999999999998
38	Niklas	Schröder	AC/DC	2.9699999999999998
35	Madalena	Sampaio	AC/DC	2.9699999999999998
44	Terhi	Hämäläinen	AC/DC	2.9699999999999998
9	Kara	Nielsen	AC/DC	1.98
34	João	Fernandes	AC/DC	1.98
57	Luis	Rojas	AC/DC	1.98
27	Patrick	Gray	AC/DC	1.98
20	Dan	Miller	AC/DC	1.98
30	Edward	Francis	AC/DC	1.98
5	František	Wichterlov	AC/DC	1.98

**Q10: 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.**

```

6
7 • WITH popular_genre AS
8 (
9     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
10    ROW_NUMBER() OVER(PARTITION BY customer.country
11    ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
12    FROM invoice_line
13    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
14    JOIN customer ON customer.customer_id = invoice.customer_id
15    JOIN track ON track.track_id = invoice_line.track_id
16    JOIN genre ON genre.genre_id = track.genre_id
17    GROUP BY 2,3,4
18    ORDER BY 2 ASC, 1 DESC
19 )
20 SELECT * FROM popular_genre WHERE RowNo <= 1;

```

**OUTPUT:**

	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	2	1

Result 3 x

Q11: 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

```
5
6 WITH Customer_with_country AS (
7     SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,
8     ROW_NUMBER() OVER(PARTITION BY billing_country ORDER BY SUM(total) DESC) AS RowNo
9     FROM invoice
10    JOIN customer ON customer.customer_id = invoice.customer_id
11    GROUP BY 1,2,3,4
12    ORDER BY 4 ASC,5 DESC)
13 SELECT * FROM Customer_with_country WHERE RowNo <= 1;
```

OUTPUT:

customer_id	first_name	last_name	billing_country	total_spending	RowNo
56	Diego	Gutiérrez	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	Luís	Gonçalves	Brazil	108.89999999999998	1
3	François	Tremblay	Canada	99.99	1
57	Luis	Rojas	Chile	97.02000000000001	1
5	František	Wichterlov	Czech Republic	144.54000000000002	1
9	Kara	Nielsen	Denmark	37.61999999999999	1
44	Terhi	Hämäläinen	Finland	79.2	1
42	Wyatt	Girard	France	99.99	1
37	Fynn	Zimmermann	Germany	94.05000000000001	1
45	Ladislav	Kovács	Hungary	78.21	1
58	Manoj	Pareek	India	111.86999999999999	1
46	Hugh	O'Reilly	Ireland	114.83999999999997	1
47	Lucas	Mancini	Italy	50.49	1
48	Johannes	Van der Berg	Netherlands	65.34	1
4	Riikka	Hanen	Norway	72.27000000000001	1