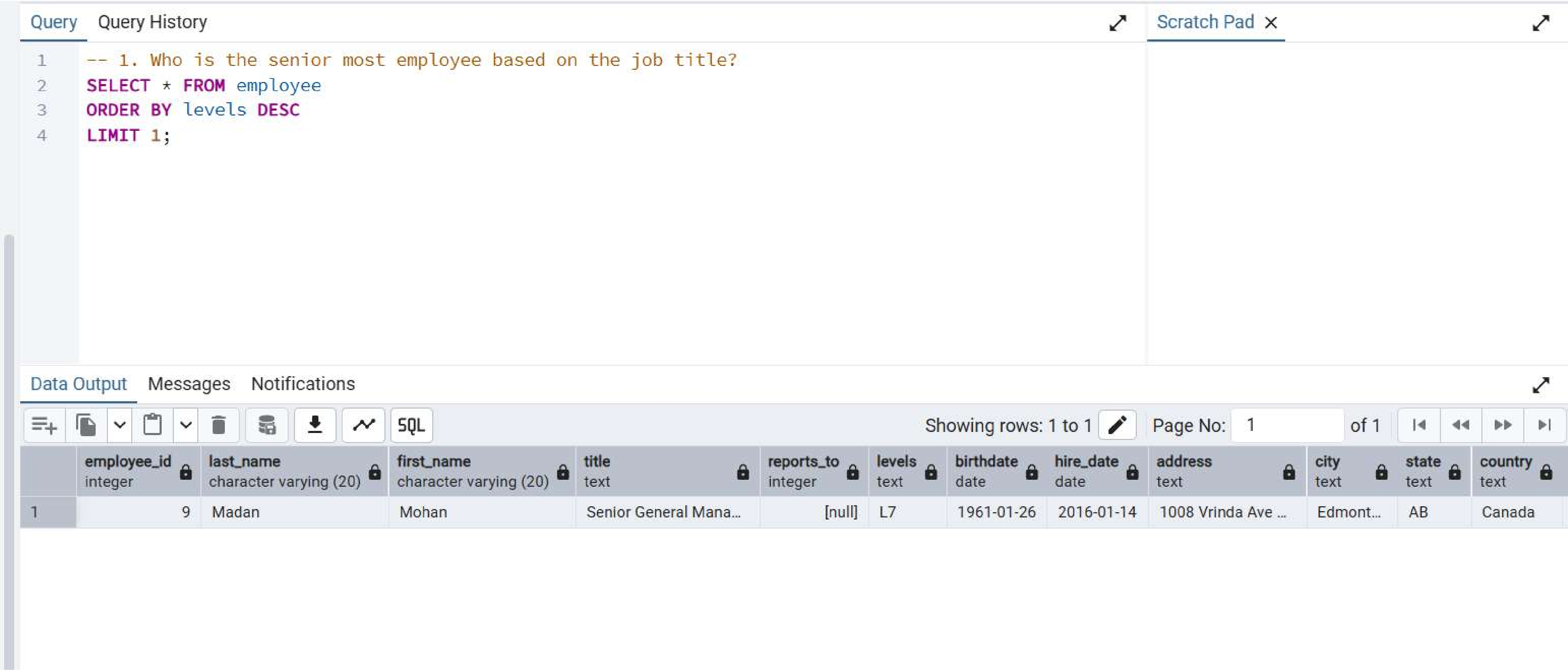
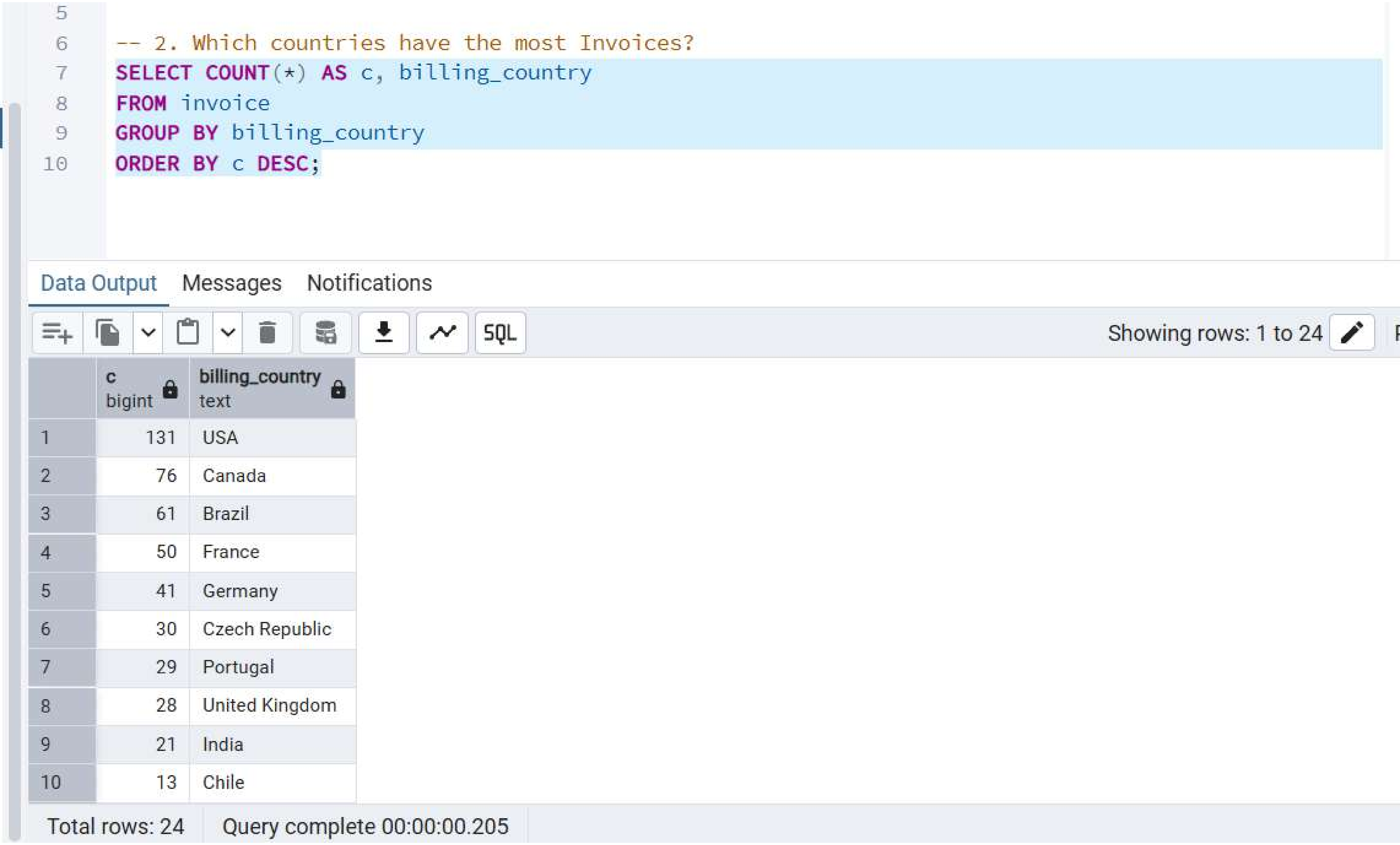


```sql
-- 1. Who is the senior most employee based on the job title?
SELECT * FROM employee
ORDER BY levels DESC
LIMIT 1;
```

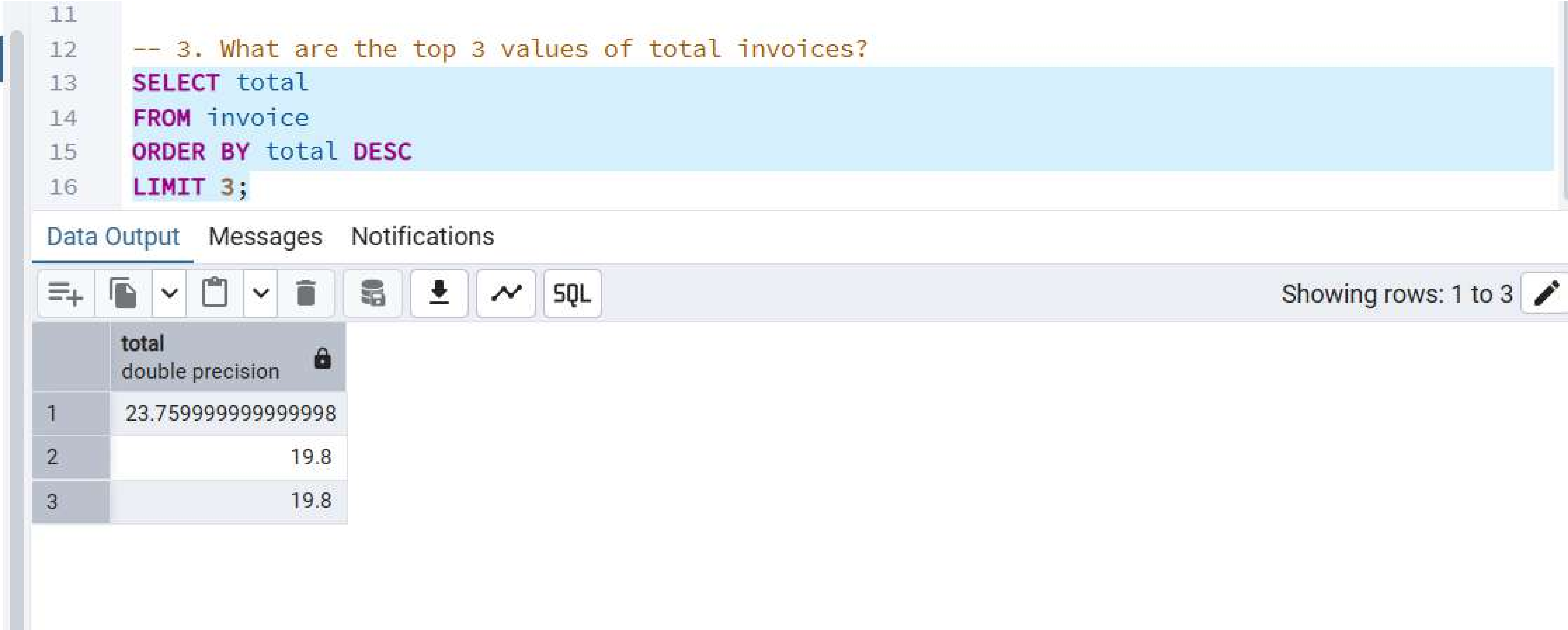
	employee_id integer	last_name character varying (20)	first_name character varying (20)	title text	reports_to integer	levels text	birthdate date	hire_date date	address text	city text	state text	country text
1	9	Madan	Mohan	Senior General Mana...	[null]	L7	1961-01-26	2016-01-14	1008 Vrinda Ave ...	Edmont...	AB	Canada



```sql
-- 2. Which countries have the most Invoices?
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC;
```

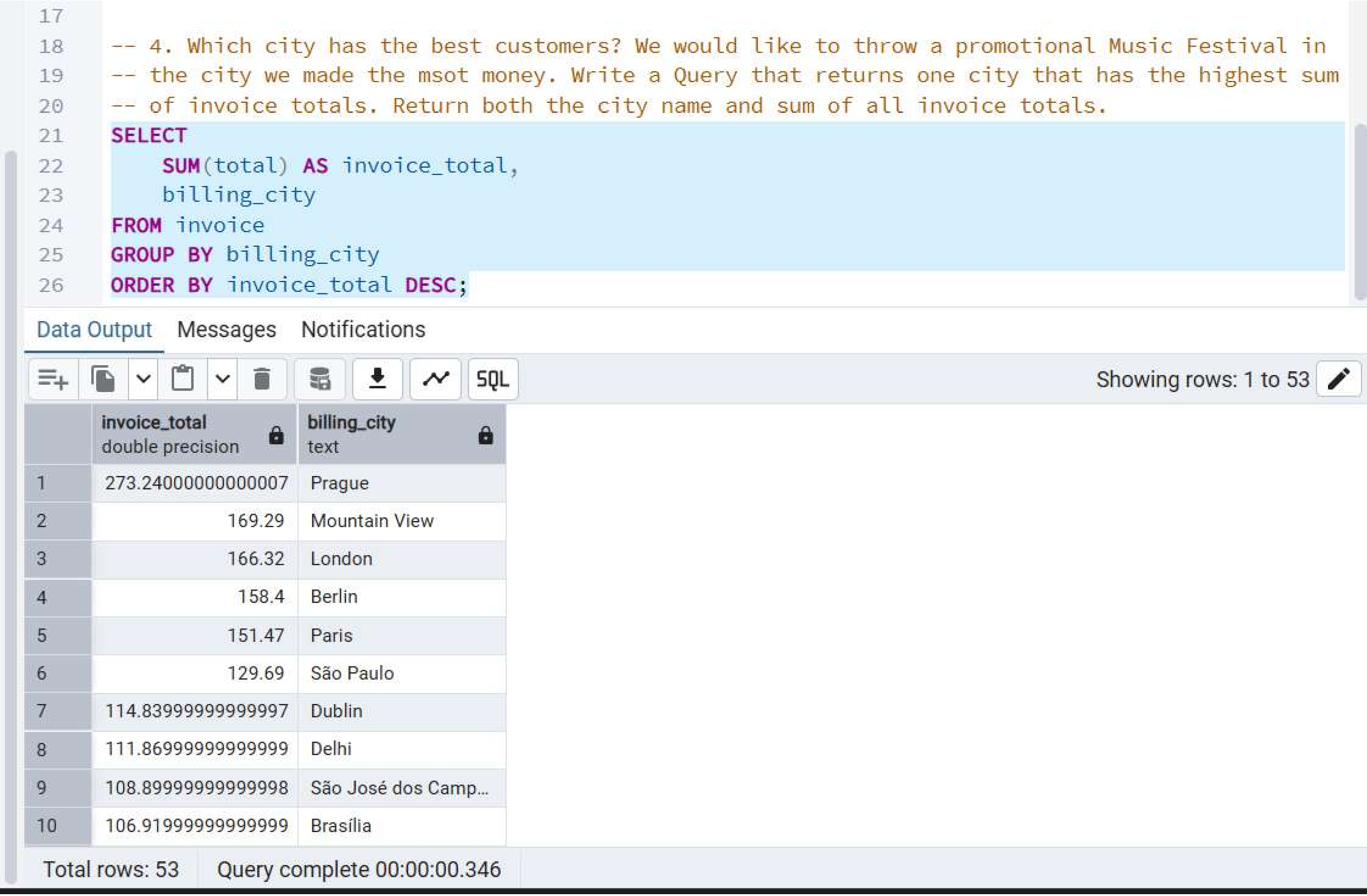
	c bigint	billing_country text
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile

Total rows: 24 Query complete 00:00:00.205



```sql
-- 3. What are the top 3 values of total invoices?
SELECT total
FROM invoice
ORDER BY total DESC
LIMIT 3;
```

	total double precision
1	23.759999999999998
2	19.8
3	19.8



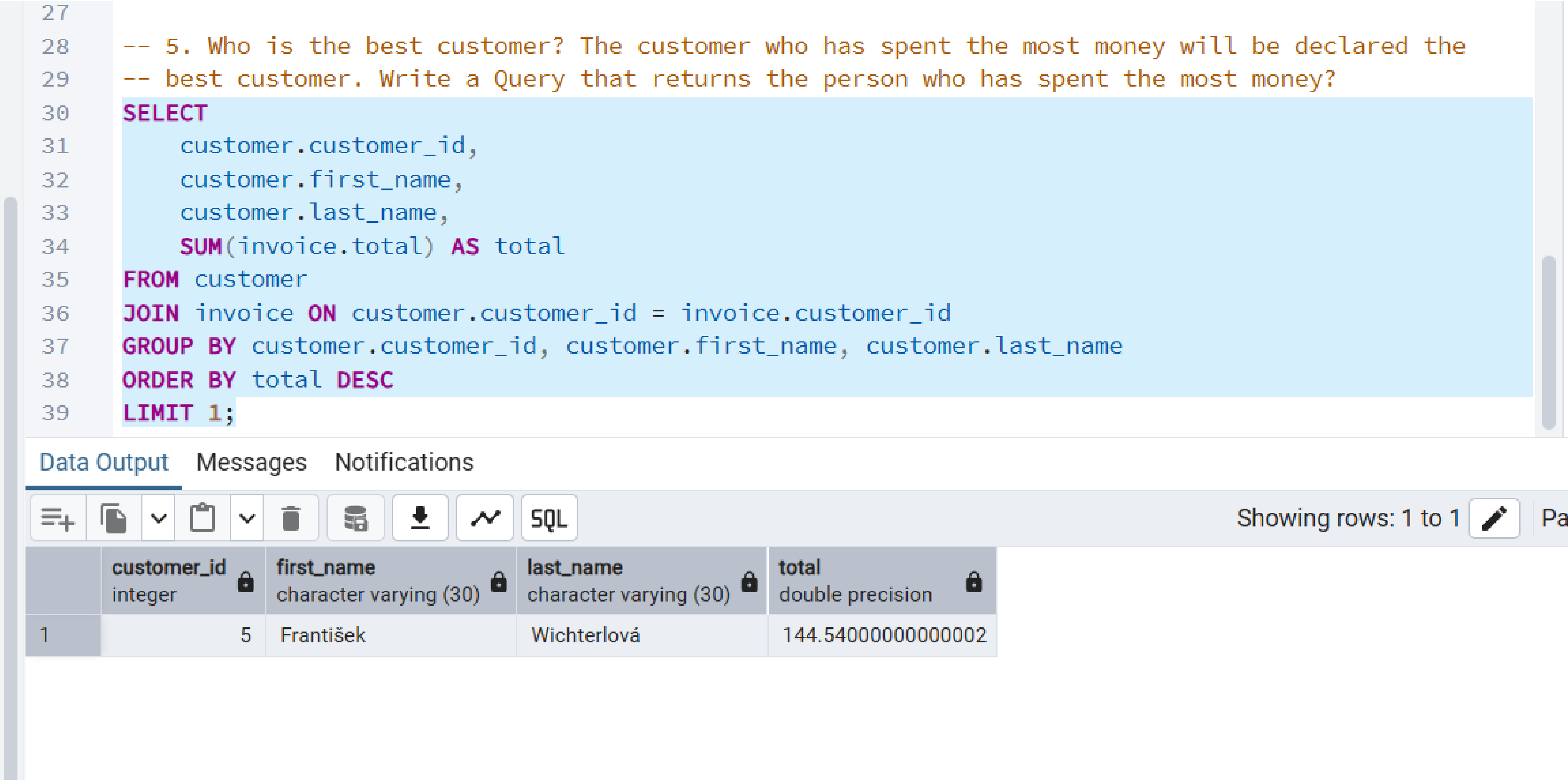
```sql

-- 4. Which city has the best customers? We would like to throw a promotional Music Festival in
-- the city we made the msot money. Write a Query that returns one city that has the highest sum
-- of invoice totals. Return both the city name and sum of all invoice totals.
SELECT
    SUM(total) AS invoice_total,
    billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC;
```

Showing rows: 1 to 53

	invoice_total double precision	billing_city text
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.83999999999997	Dublin
8	111.86999999999999	Delhi
9	108.89999999999998	São José dos Camp...
10	106.91999999999999	Brasília

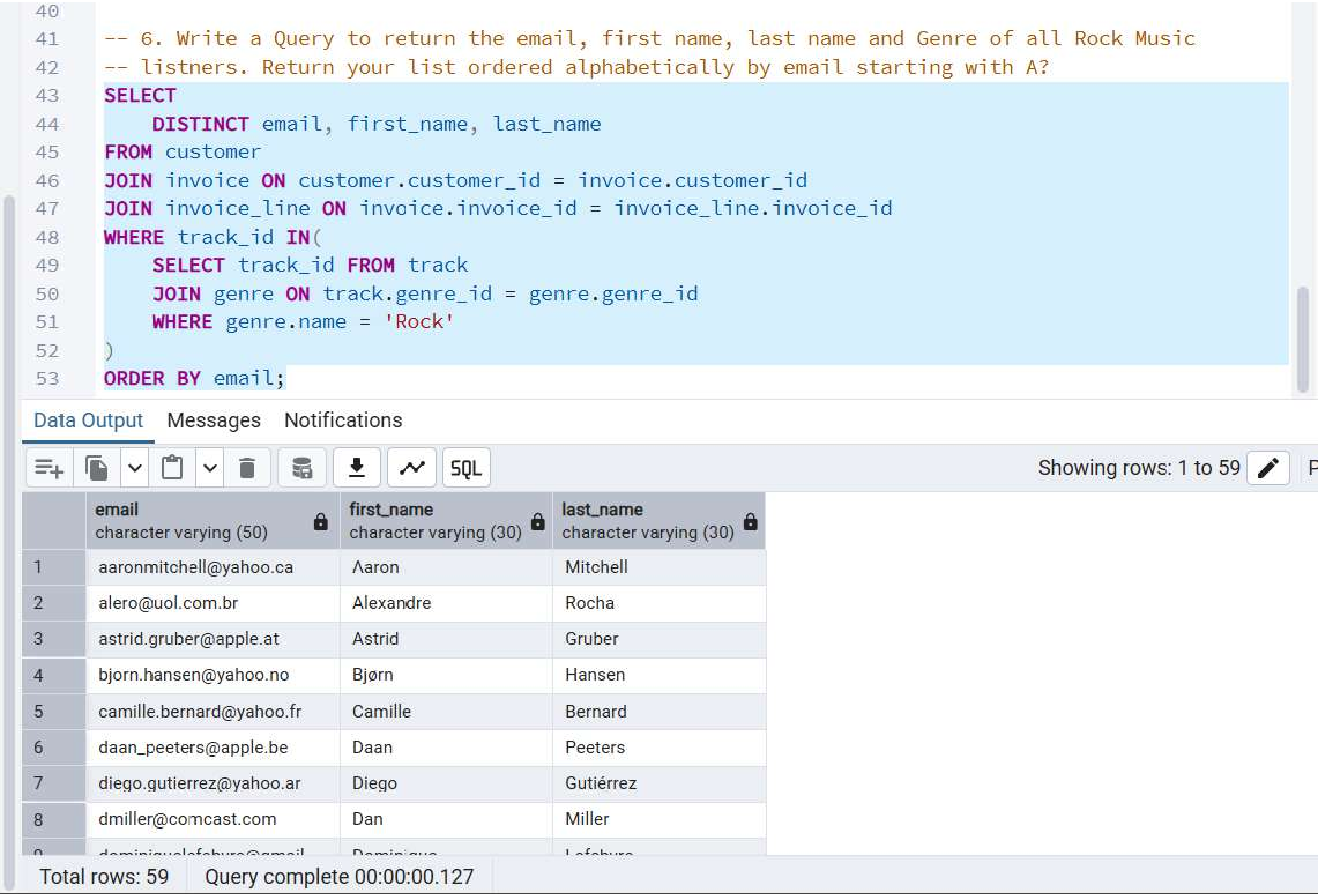
Total rows: 53 Query complete 00:00:00.346



```

-- 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?
SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    SUM(invoice.total) AS total
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id, customer.first_name, customer.last_name
ORDER BY total DESC
LIMIT 1;
```

	customer_id integer	first_name character varying (30)	last_name character varying (30)	total double precision
1	5	František	Wichterlová	144.54000000000002

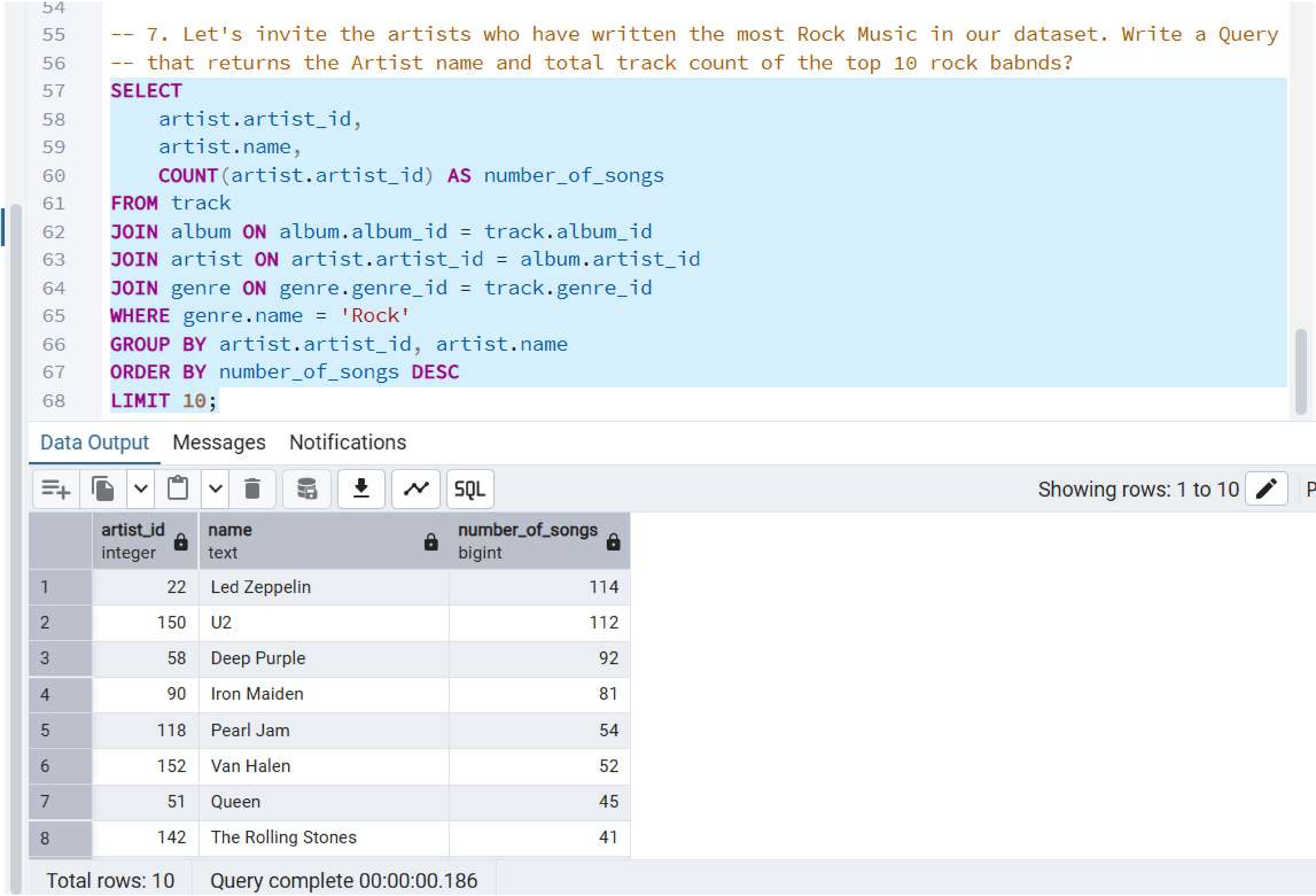


```sql

-- 6. Write a Query to return the email, first name, last name and Genre of all Rock Music
-- listners. Return your list ordered alphabetically by email starting with A?
SELECT
    DISTINCT email, first_name, last_name
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
WHERE track_id IN(
    SELECT track_id FROM track
    JOIN genre ON track.genre_id = genre.genre_id
    WHERE genre.name = 'Rock'
)
ORDER BY email;
```

	email character varying (50)	first_name character varying (30)	last_name character varying (30)
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	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail	Dominique	Lefebvre

Total rows: 59 Query complete 00:00:00.127

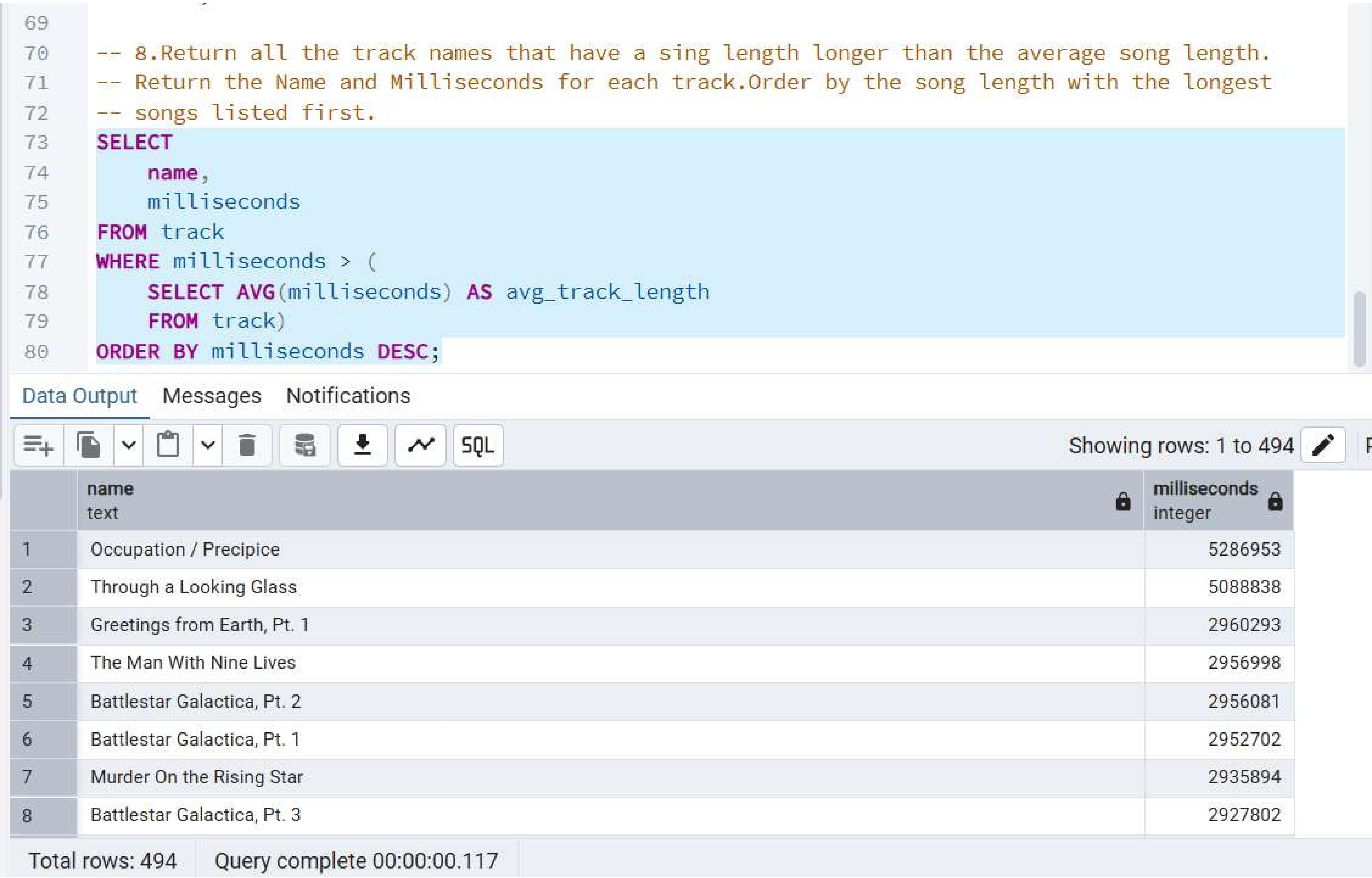


```sql
-- 7. Let's invite the artists who have written the most Rock Music in our dataset. Write a Query
-- that returns the Artist name and total track count of the top 10 rock babnds?
SELECT
    artist.artist_id,
    artist.name,
    COUNT(artist.artist_id) AS number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name = 'Rock'
GROUP BY artist.artist_id, artist.name
ORDER BY number_of_songs DESC
LIMIT 10;
```

Showing rows: 1 to 10

	artist_id integer	name text	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41

Total rows: 10 Query complete 00:00:00.186



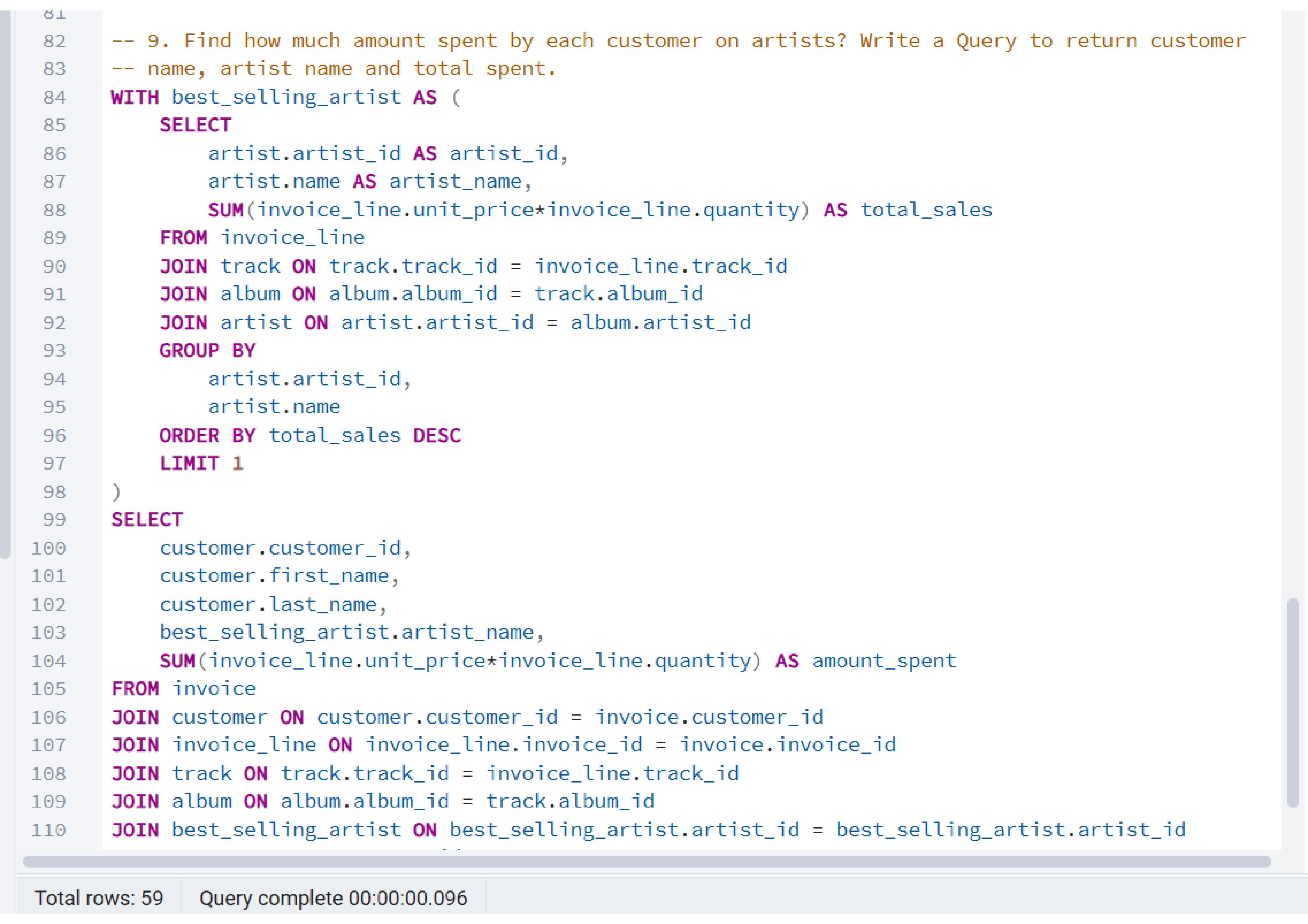
```sql

-- 8.Return all the track names that have a sing 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.
SELECT
    name,
    milliseconds
FROM track
WHERE milliseconds > (
    SELECT AVG(milliseconds) AS avg_track_length
    FROM track)
ORDER BY milliseconds DESC;
```

Showing rows: 1 to 494

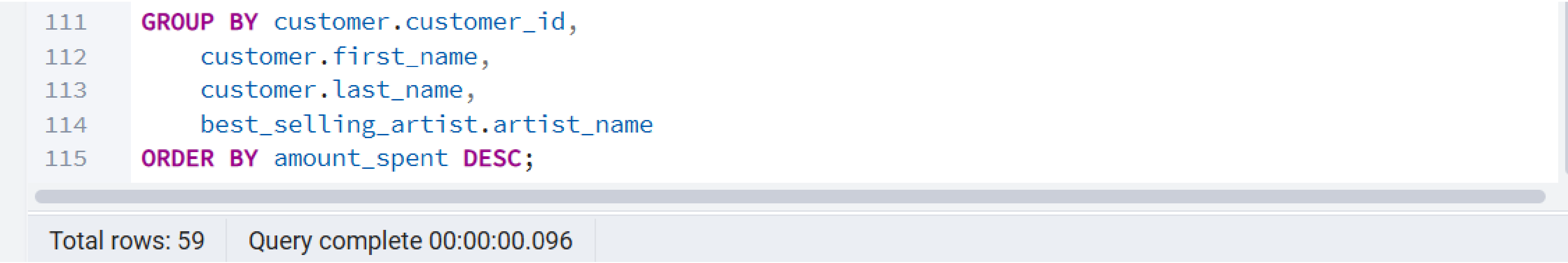
	name text	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802

Total rows: 494   Query complete 00:00:00.117



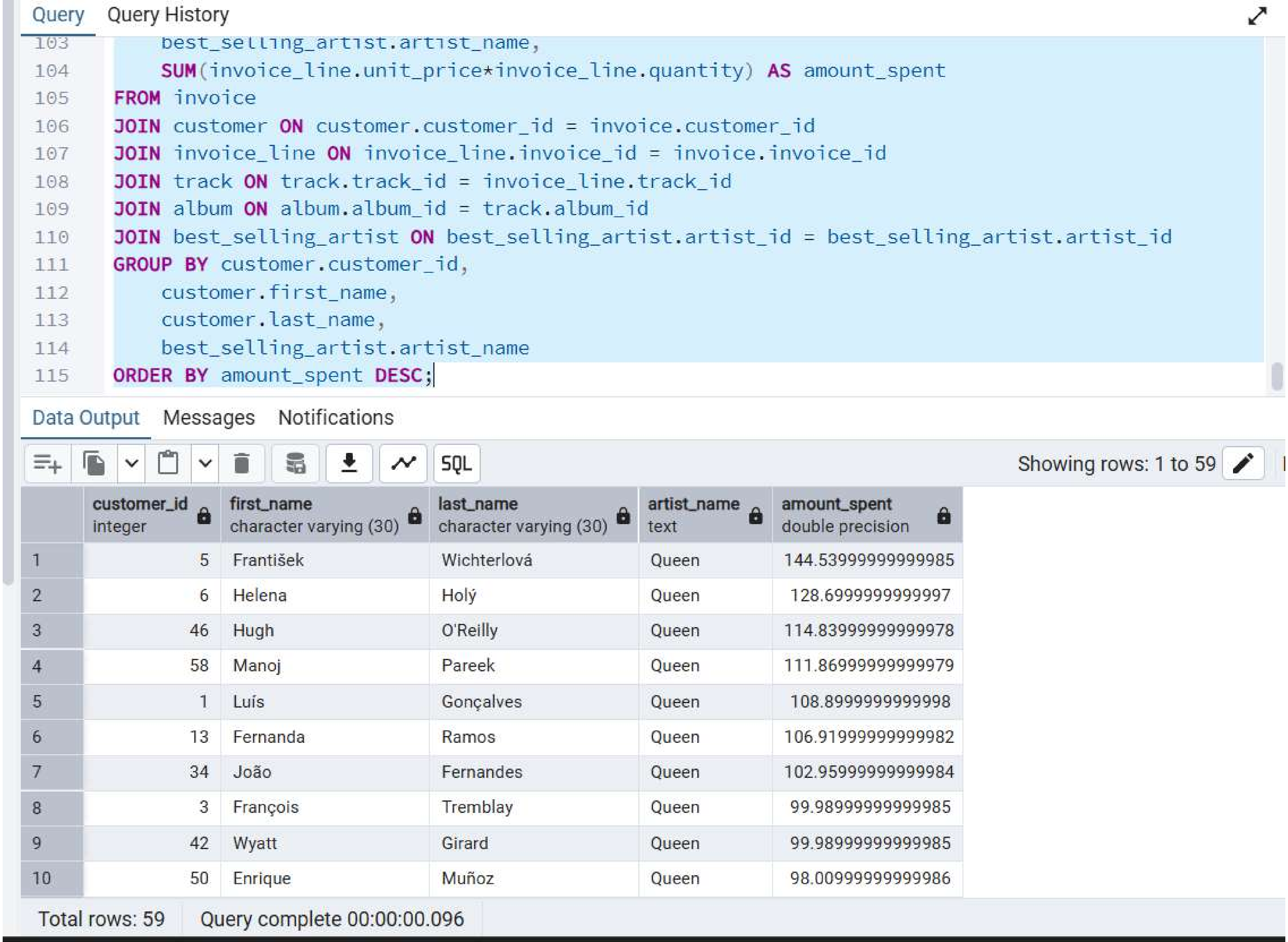
```sql
-- 9. Find how much amount spent by each customer on artists? Write a Query to return customer
-- name, artist name and total spent.
WITH best_selling_artist AS (
    SELECT
        artist.artist_id AS artist_id,
        artist.name AS artist_name,
        SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
    FROM invoice_line
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN album ON album.album_id = track.album_id
    JOIN artist ON artist.artist_id = album.artist_id
    GROUP BY
        artist.artist_id,
        artist.name
    ORDER BY total_sales DESC
    LIMIT 1
)
SELECT
    customer.customer_id,
    customer.first_name,
    customer.last_name,
    best_selling_artist.artist_name,
    SUM(invoice_line.unit_price*invoice_line.quantity) AS amount_spent
FROM invoice
JOIN customer ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN album ON album.album_id = track.album_id
JOIN best_selling_artist ON best_selling_artist.artist_id = best_selling_artist.artist_id
```

Total rows: 59   Query complete 00:00:00.096



```sql
GROUP BY customer.customer_id,
    customer.first_name,
    customer.last_name,
    best_selling_artist.artist_name
ORDER BY amount_spent DESC;
```

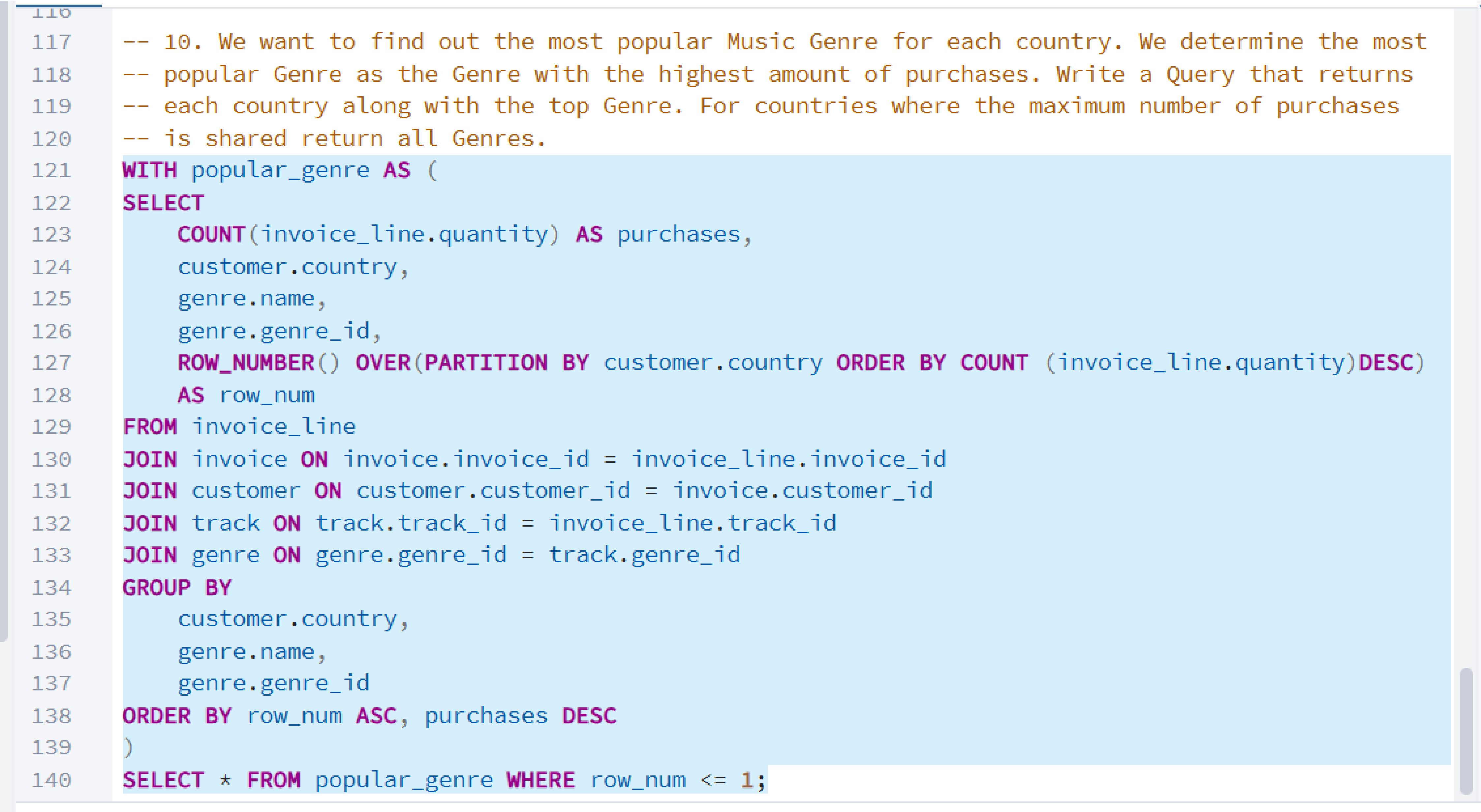
Total rows: 59   Query complete 00:00:00.096



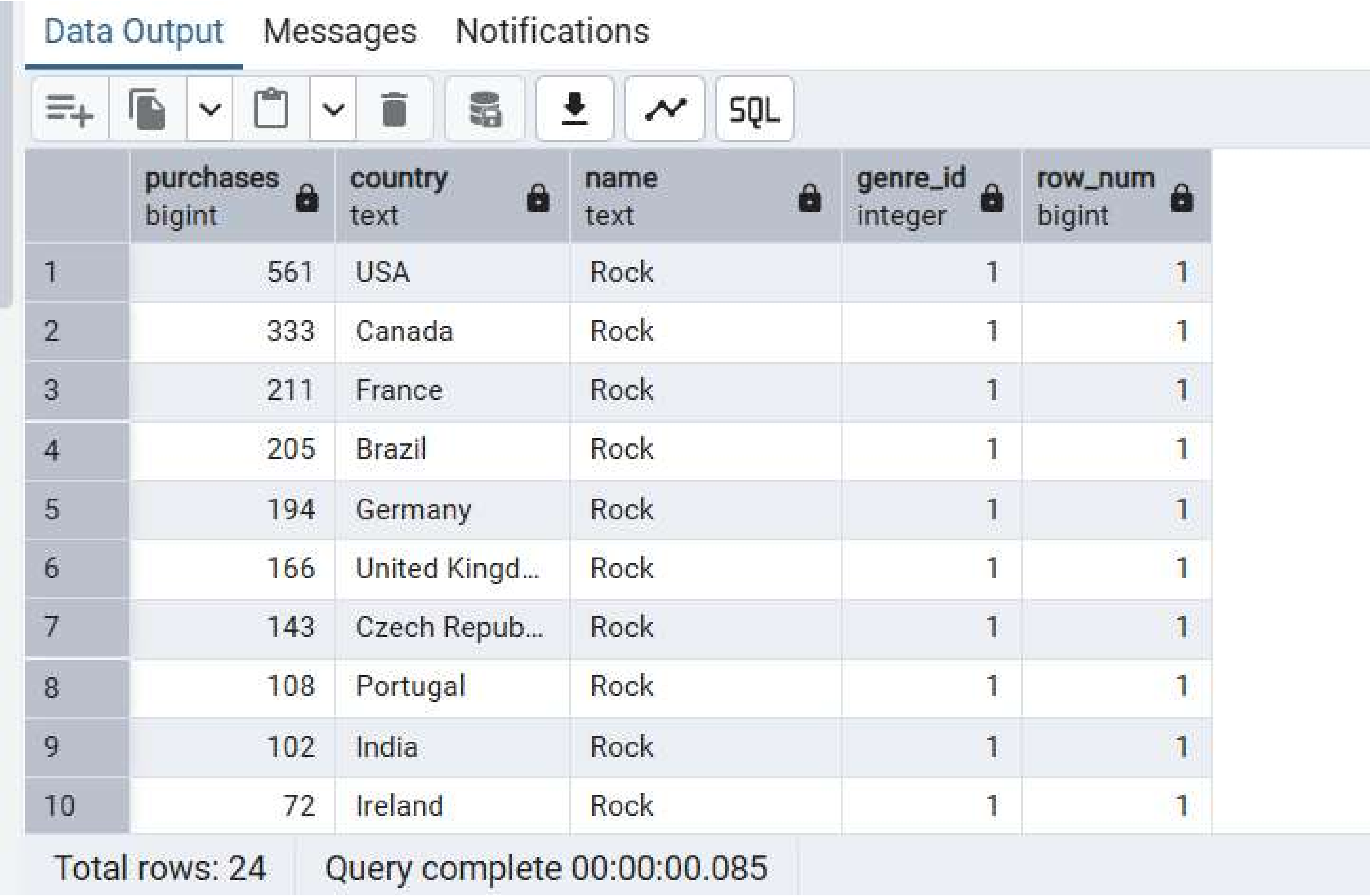
```sql
    best_selling_artist.artist_name,
    SUM(invoice_line.unit_price*invoice_line.quantity) AS amount_spent
FROM invoice
JOIN customer ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN album ON album.album_id = track.album_id
JOIN best_selling_artist ON best_selling_artist.artist_id = best_selling_artist.artist_id
GROUP BY customer.customer_id,
    customer.first_name,
    customer.last_name,
    best_selling_artist.artist_name
ORDER BY amount_spent DESC;
```

	customer_id integer	first_name character varying (30)	last_name character varying (30)	artist_name text	amount_spent double precision
1	5	František	Wichterlová	Queen	144.53999999999985
2	6	Helena	Holý	Queen	128.6999999999997
3	46	Hugh	O'Reilly	Queen	114.83999999999978
4	58	Manoj	Pareek	Queen	111.86999999999979
5	1	Luís	Gonçalves	Queen	108.8999999999998
6	13	Fernanda	Ramos	Queen	106.91999999999982
7	34	João	Fernandes	Queen	102.95999999999984
8	3	François	Tremblay	Queen	99.98999999999985
9	42	Wyatt	Girard	Queen	99.98999999999985
10	50	Enrique	Muñoz	Queen	98.00999999999986

Total rows: 59 Query complete 00:00:00.096

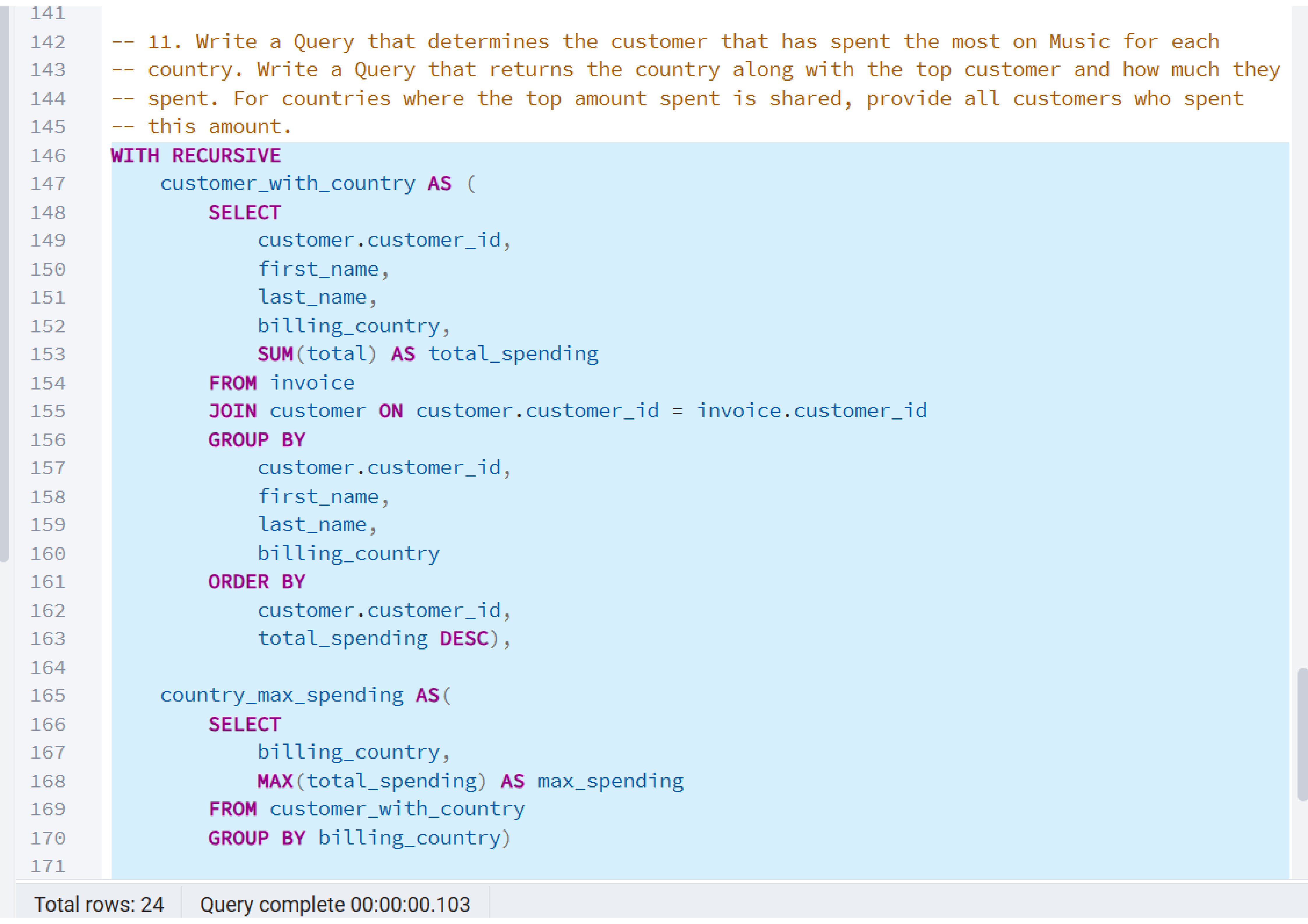


```sql
-- 10. We want to find out the most popular Music Genre for each country. We determine the most
-- popular Genre as the Genre with the highest amount of purchases. Write a Query that returns
-- each country along with the top Genre. For countries where the maximum number of purchases
-- is shared return all Genres.
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 row_num
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
    customer.country,
    genre.name,
    genre.genre_id
ORDER BY row_num ASC, purchases DESC
)
SELECT * FROM popular_genre WHERE row_num <= 1;
```



	purchases bigint	country text	name text	genre_id integer	row_num bigint
1	561	USA	Rock	1	1
2	333	Canada	Rock	1	1
3	211	France	Rock	1	1
4	205	Brazil	Rock	1	1
5	194	Germany	Rock	1	1
6	166	United Kingd...	Rock	1	1
7	143	Czech Repub...	Rock	1	1
8	108	Portugal	Rock	1	1
9	102	India	Rock	1	1
10	72	Ireland	Rock	1	1

Total rows: 24 Query complete 00:00:00.085



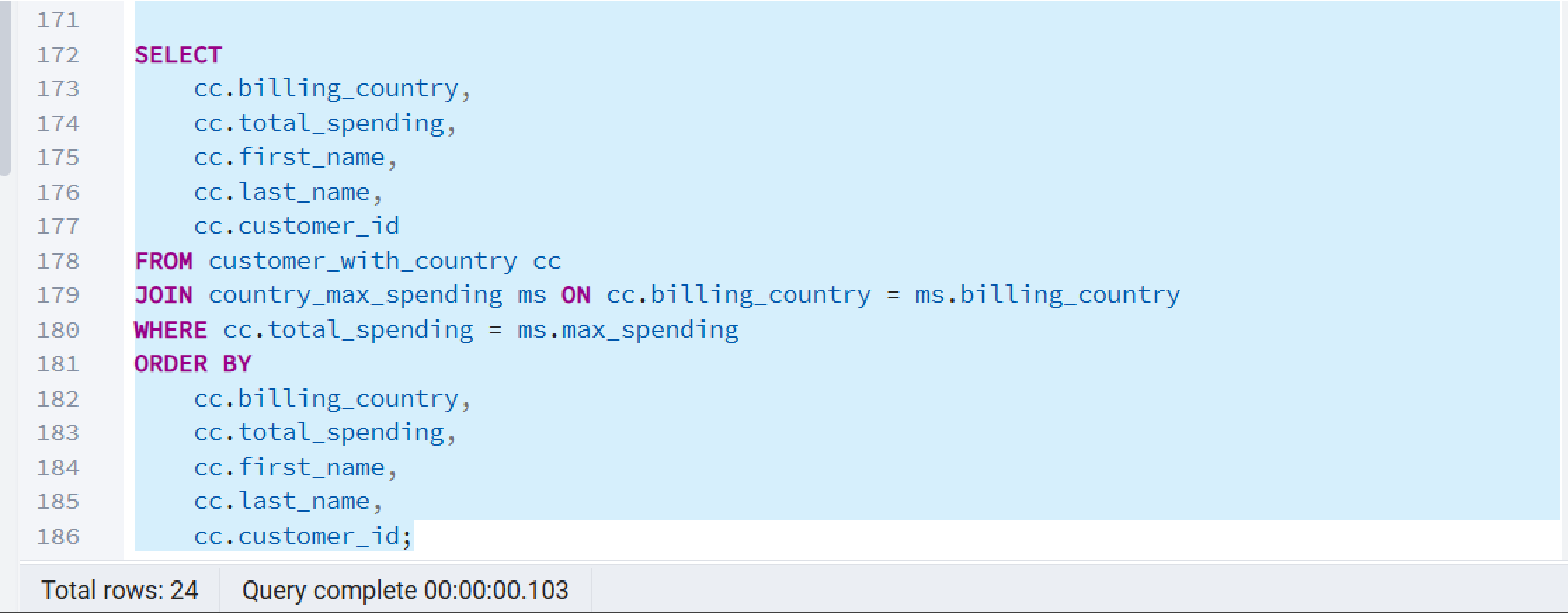
```sql

-- 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.
WITH RECURSIVE
    customer_with_country AS (
        SELECT
            customer.customer_id,
            first_name,
            last_name,
            billing_country,
            SUM(total) AS total_spending
        FROM invoice
        JOIN customer ON customer.customer_id = invoice.customer_id
        GROUP BY
            customer.customer_id,
            first_name,
            last_name,
            billing_country
        ORDER BY
            customer.customer_id,
            total_spending DESC),

    country_max_spending AS(
        SELECT
            billing_country,
            MAX(total_spending) AS max_spending
        FROM customer_with_country
        GROUP BY billing_country)

```

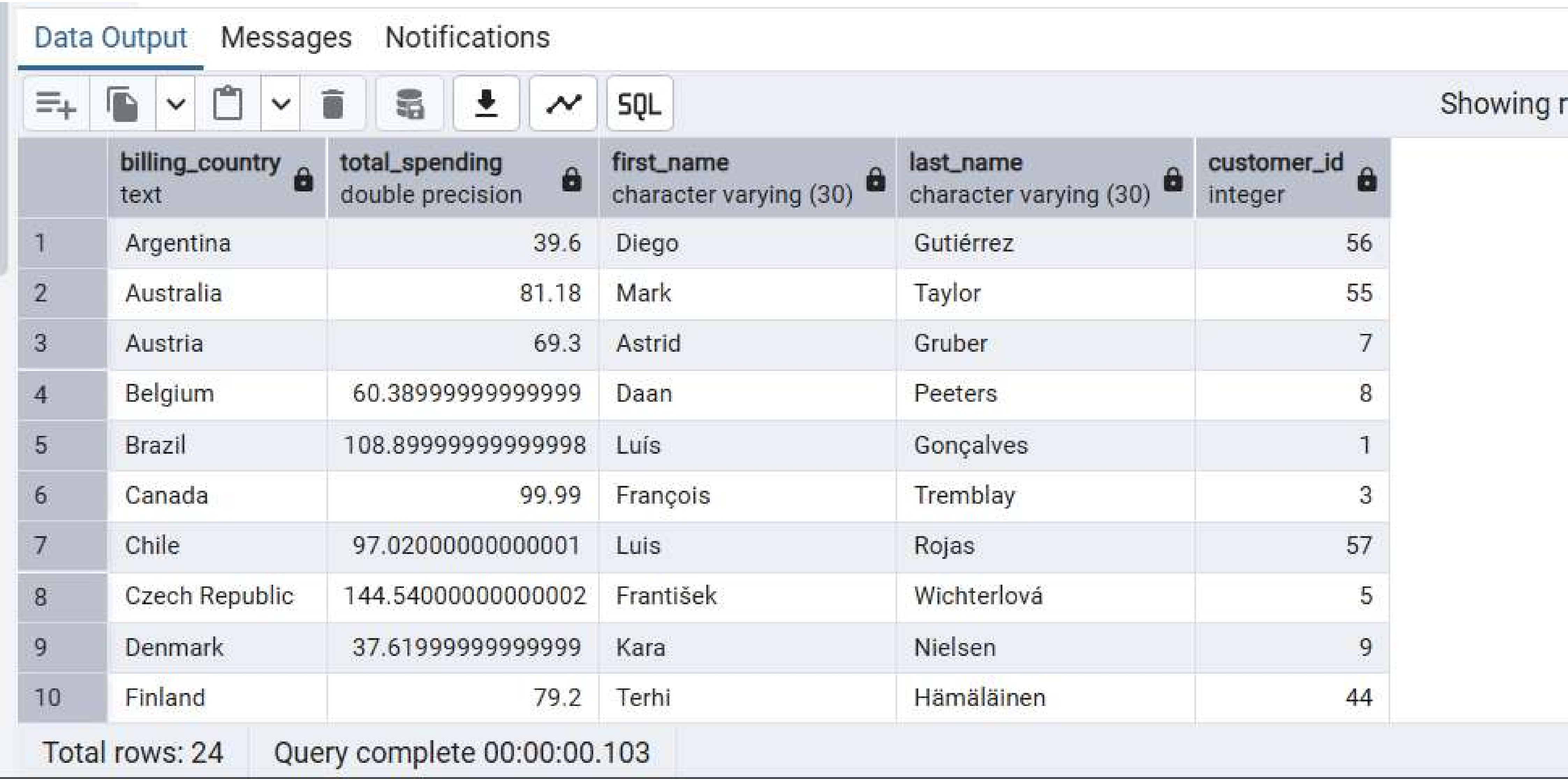
Total rows: 24   Query complete 00:00:00.103



```sql

SELECT
    cc.billing_country,
    cc.total_spending,
    cc.first_name,
    cc.last_name,
    cc.customer_id
FROM customer_with_country cc
JOIN country_max_spending ms ON cc.billing_country = ms.billing_country
WHERE cc.total_spending = ms.max_spending
ORDER BY
    cc.billing_country,
    cc.total_spending,
    cc.first_name,
    cc.last_name,
    cc.customer_id;
```

Total rows: 24   Query complete 00:00:00.103



	billing_country text	total_spending double precision	first_name character varying (30)	last_name character varying (30)	customer_id integer
1	Argentina	39.6	Diego	Gutiérrez	56
2	Australia	81.18	Mark	Taylor	55
3	Austria	69.3	Astrid	Gruber	7
4	Belgium	60.38999999999999	Daan	Peeters	8
5	Brazil	108.89999999999998	Luís	Gonçalves	1
6	Canada	99.99	François	Tremblay	3
7	Chile	97.02000000000001	Luis	Rojas	57
8	Czech Republic	144.54000000000002	František	Wichterlová	5
9	Denmark	37.61999999999999	Kara	Nielsen	9
10	Finland	79.2	Terhi	Hämäläinen	44

Total rows: 24 Query complete 00:00:00.103