SQL PROJECT

MUSIC STORE ANALYSIS

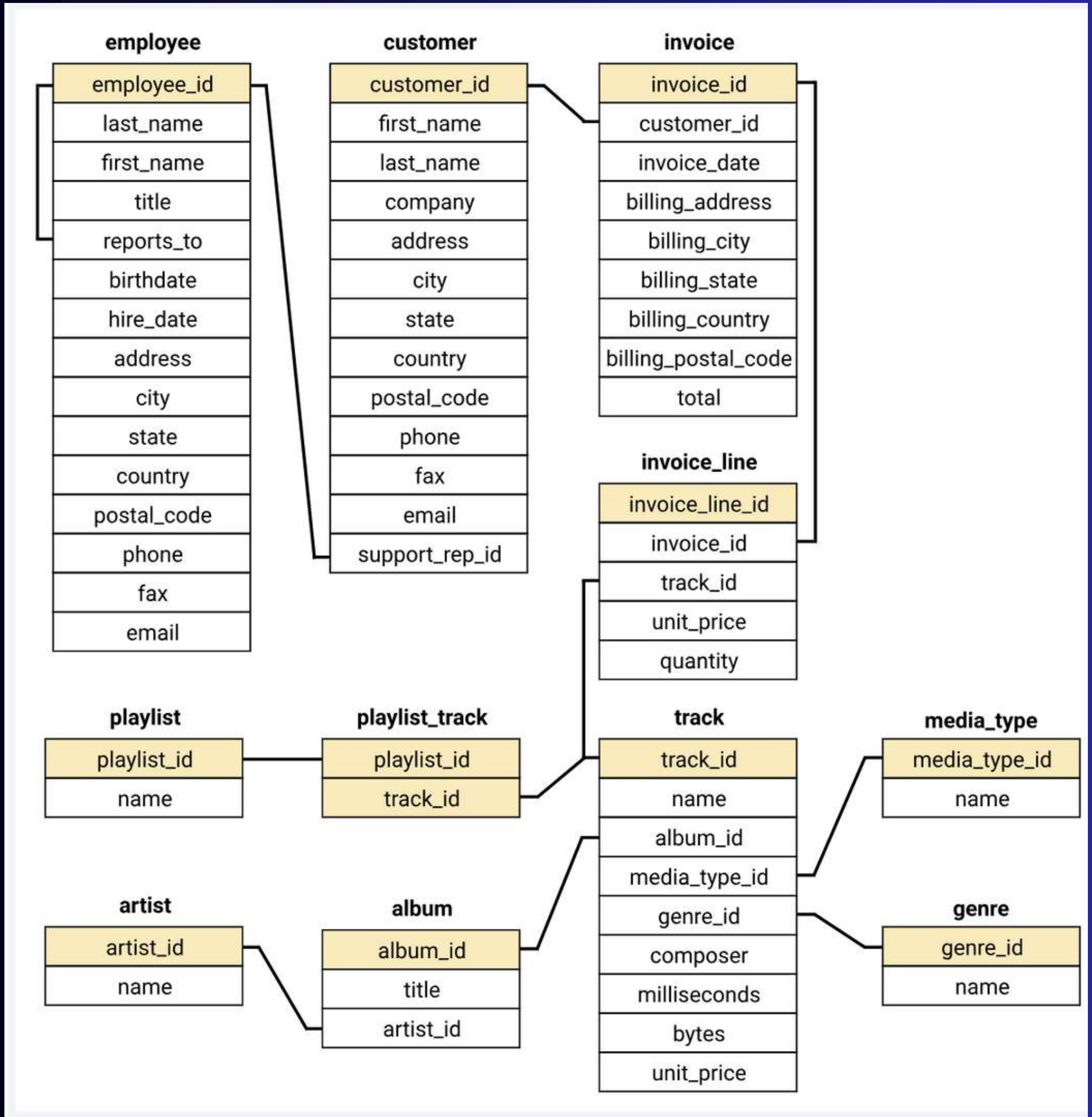
-Pavan Kumar

OBJECTIVE

- The primary objective of the music store is to achieve sustainable business growth while addressing existing challenges.

- We need to examine the dataset with SQL and help the music store understand its business growth by answering simple questions.

MUSIC PLAYLIST DATABASE SCHEMA



employee
employee_id
last_name
first_name
title
reports_to
birthdate
hire_date
address
city
state
country
postal_code
phone
fax
email

customer
customer_id
first_name
last_name
company
address
city
state
country
postal_code
phone
fax
email
support_rep_id

invoice
invoice_id
customer_id
invoice_date
billing_address
billing_city
billing_state
billing_country
billing_postal_code
total

invoice_line
invoice_line_id
invoice_id
track_id
unit_price
quantity

playlist
playlist_id
name

playlist_track
playlist_id
track_id

track
track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price

media_type
media_type_id
name

artist
artist_id
name

album
album_id
title
artist_id

genre
genre_id
name

-Pavan Kumar

LEVEL OF QUERIES

EASY

Queries include:

SELECT, GROUP BY, ORDER BY, LIMIT, DESC

MODERATE

Queries include:

JOINS, GROUP BY, ORDER BY, LIMIT

ADVANCE

Queries include:

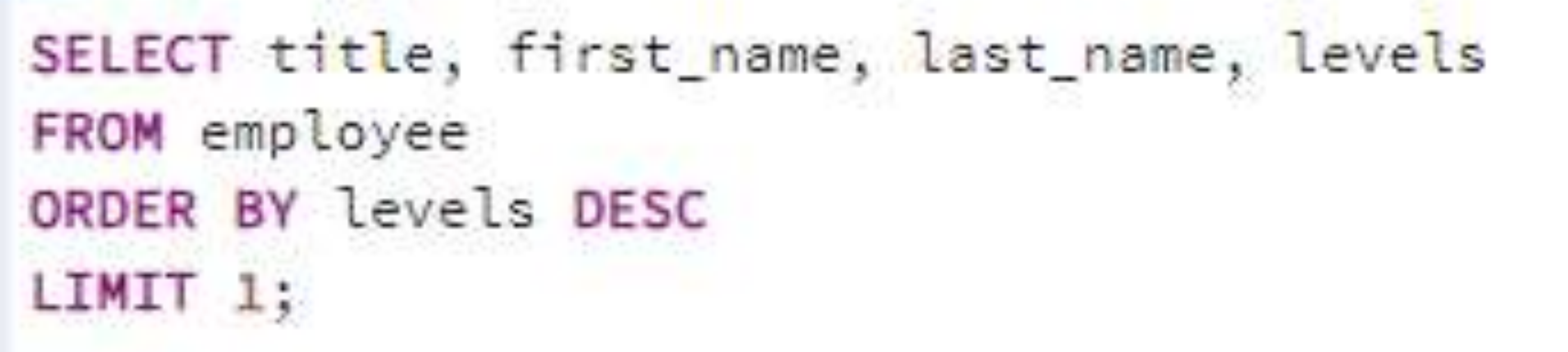
CTE (COMMON TABLE EXPRESSION)

EASY

QUESTION-1

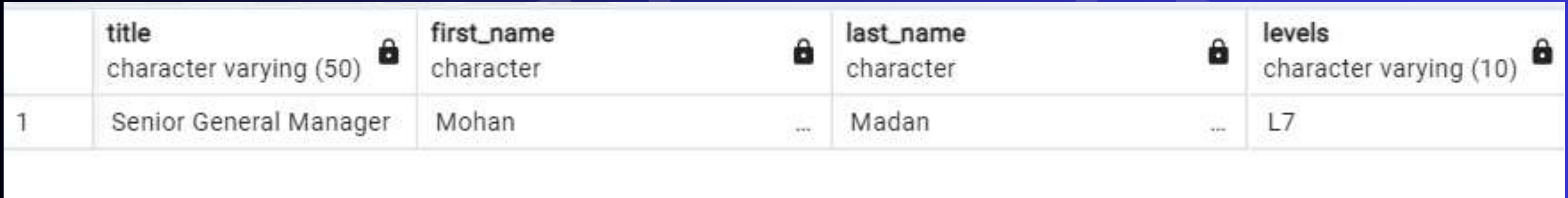
Who is the senior most employee based on job title?

QUERY:



```
SELECT title, first_name, last_name, levels
FROM employee
ORDER BY levels DESC
LIMIT 1;
```

OUTPUT:



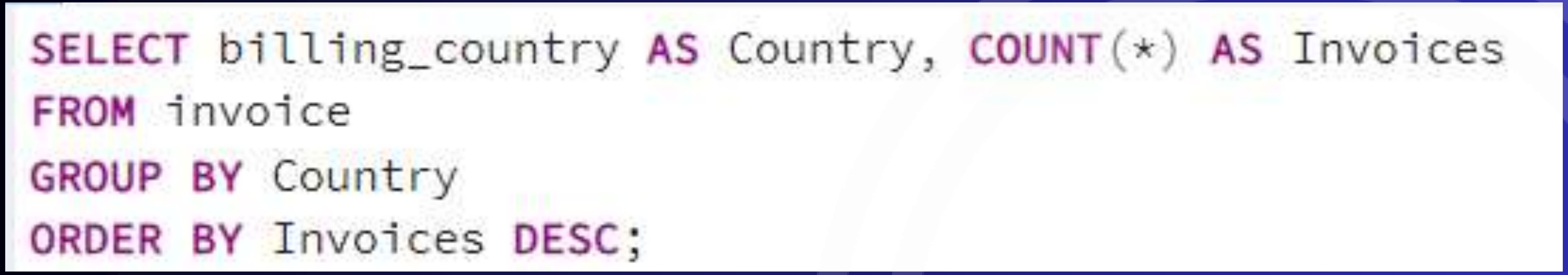
	title character varying (50)	first_name character	last_name character	levels character varying (10)
1	Senior General Manager	Mohan	Madan	L7

EASY

QUESTION-2

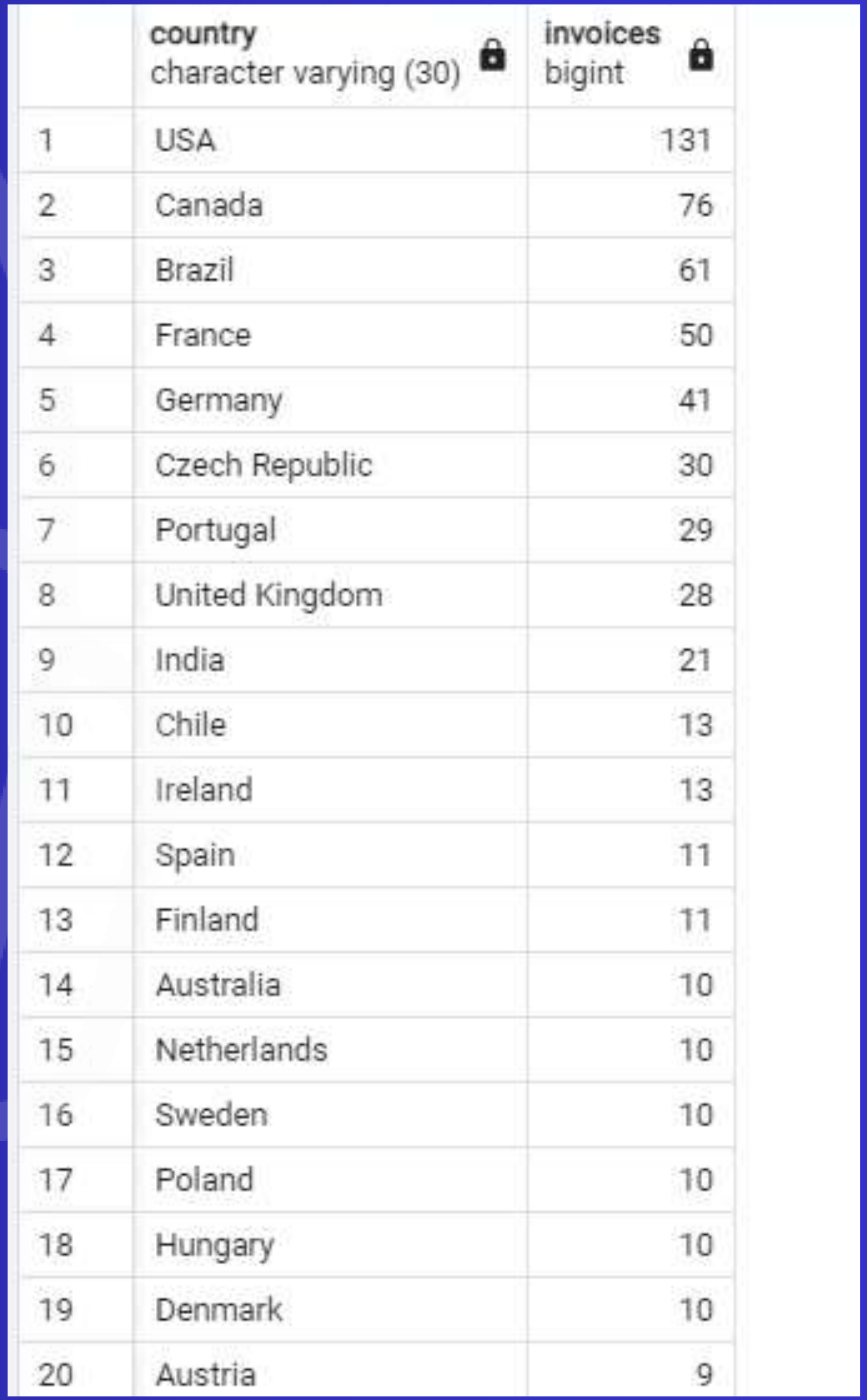
Which countries have the most Invoices?

QUERY:



```
SELECT billing_country AS Country, COUNT(*) AS Invoices
FROM invoice
GROUP BY Country
ORDER BY Invoices DESC;
```

OUTPUT:



	country character varying (30)	invoices bigint
1	USA	131
2	Canada	76
3	Brazil	61
4	France	50
5	Germany	41
6	Czech Republic	30
7	Portugal	29
8	United Kingdom	28
9	India	21
10	Chile	13
11	Ireland	13
12	Spain	11
13	Finland	11
14	Australia	10
15	Netherlands	10
16	Sweden	10
17	Poland	10
18	Hungary	10
19	Denmark	10
20	Austria	9

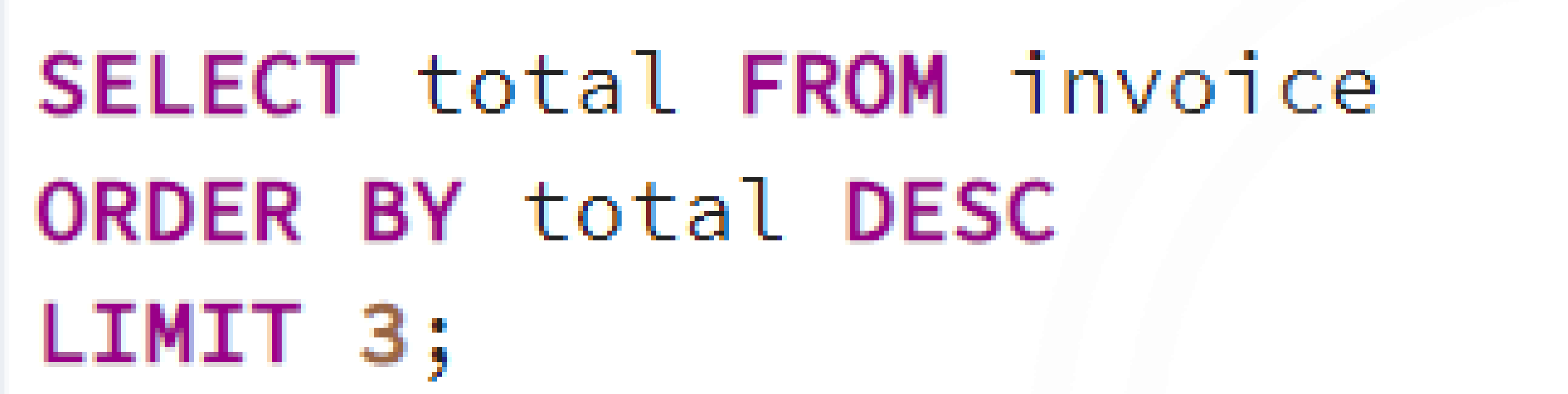
-Pavan Kumar

EASY

QUESTION-3

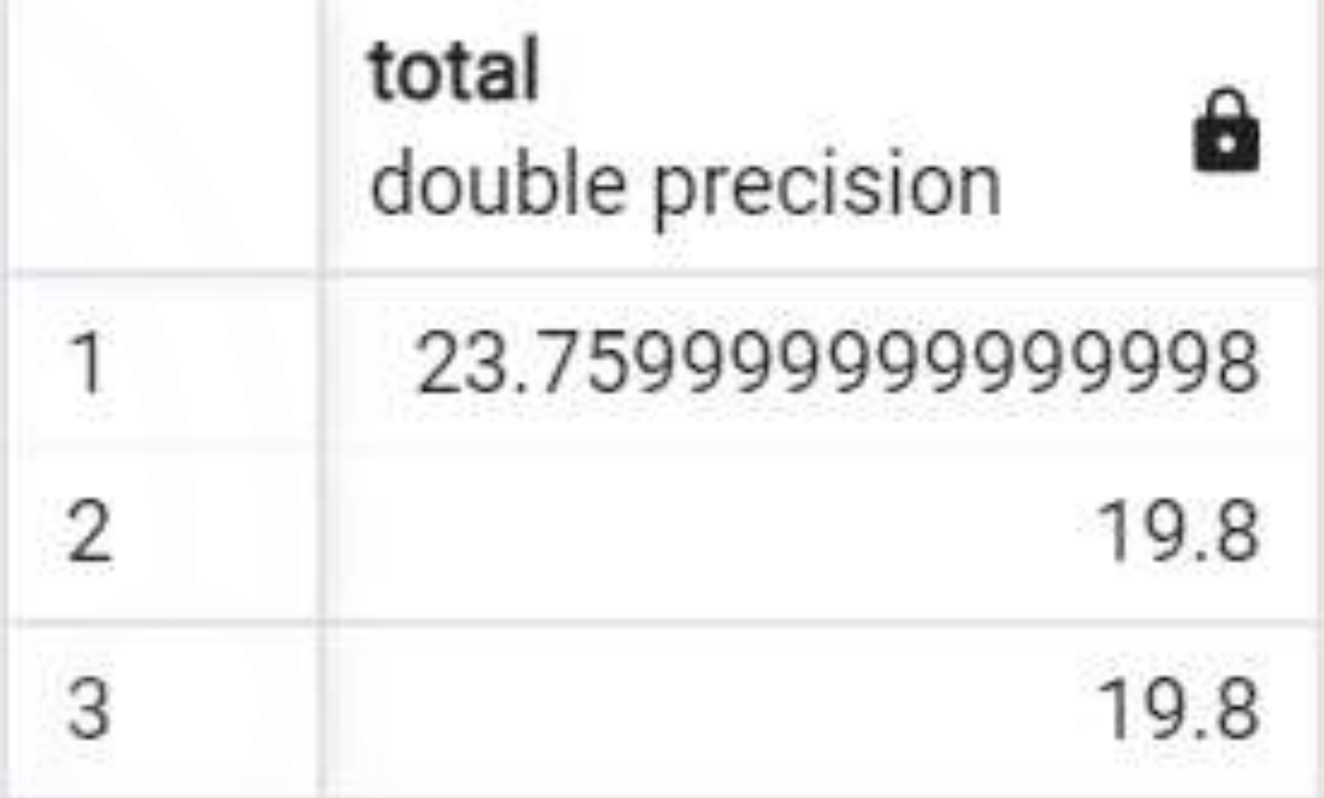
What are top 3 values of total invoice?

QUERY:



```sql
SELECT total FROM invoice
ORDER BY total DESC
LIMIT 3;
```

OUTPUT:



	total double precision
1	23.759999999999998
2	19.8
3	19.8

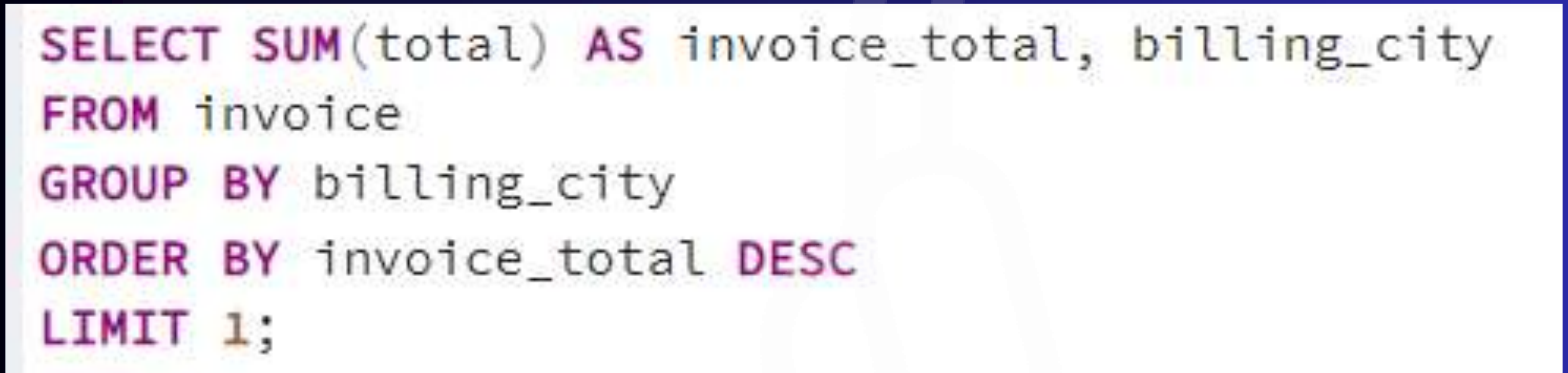
EASY

QUESTION-4

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.

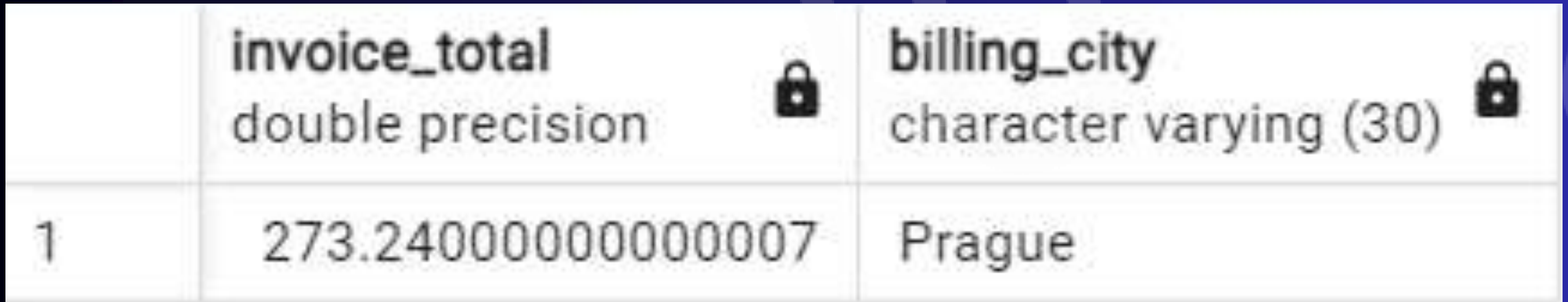
Return both the city name & sum of all invoice totals.

QUERY:



```
SELECT SUM(total) AS invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC
LIMIT 1;
```

OUTPUT:



	invoice_total double precision 🔒	billing_city character varying (30) 🔒
1	273.24000000000007	Prague

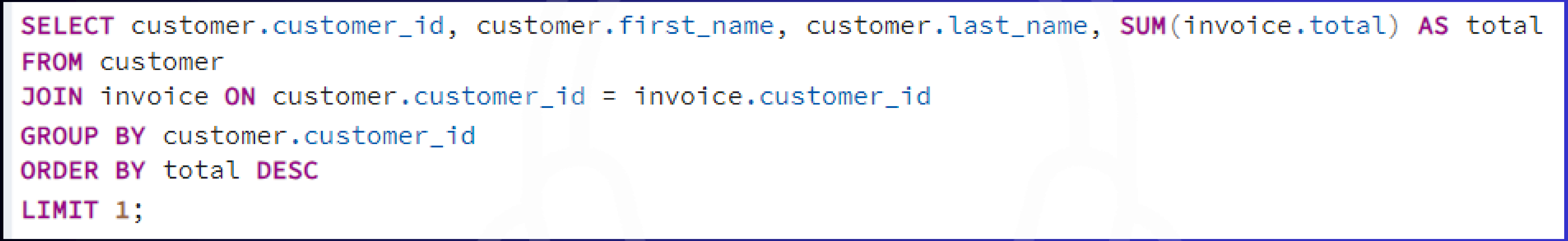
-Pavan Kumar

QUESTION-5

Who is the best customer? The customer who has spent the most money will be declared the best customer.

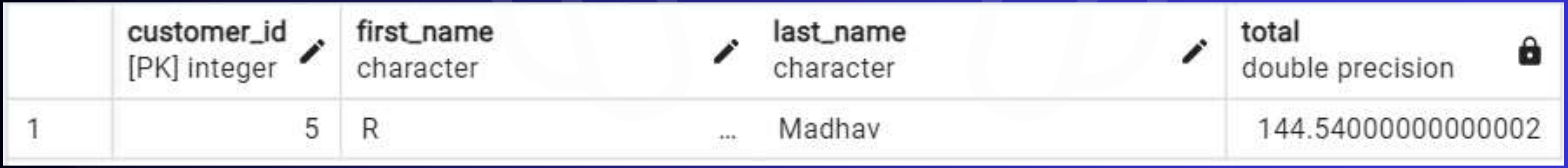
Write a query that returns the person who has spent the most money.

QUERY:



```
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
ORDER BY total DESC
LIMIT 1;
```

OUTPUT:



	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R ...	Madhav	144.5400000000002

-Pavan Kumar

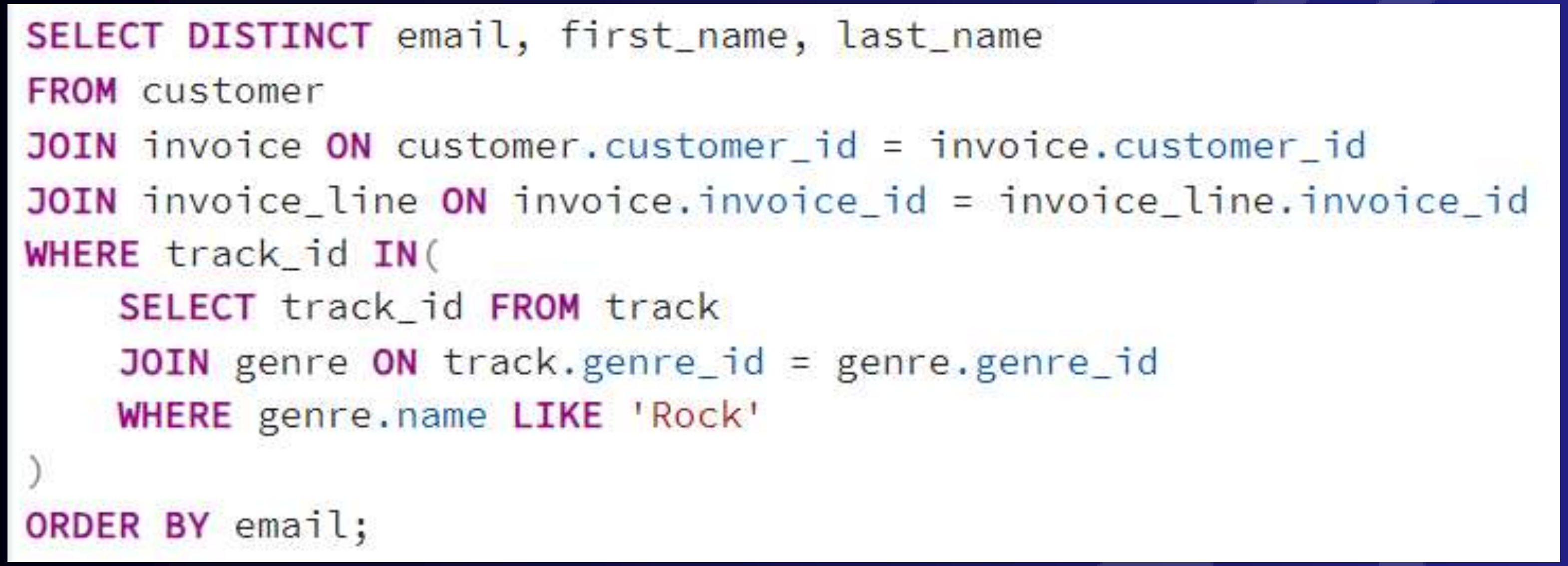
MODERATE

QUESTION-1

Write query to return the email, first name, last name, & Genre of all Rock Music listeners.

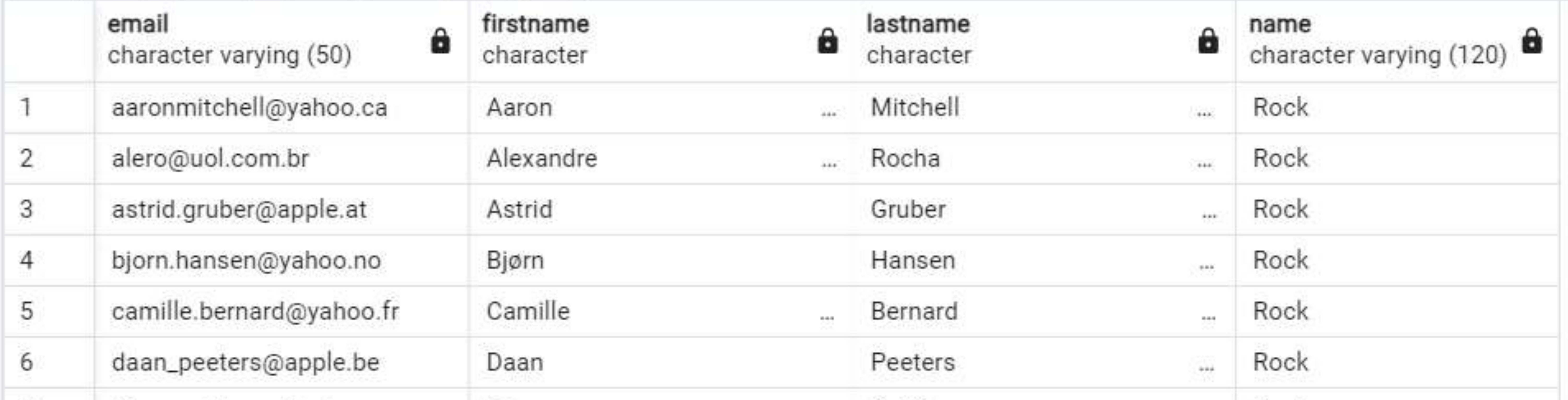
Return your list ordered alphabetically by email starting with A.

QUERY:



```
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 LIKE 'Rock'
)
ORDER BY email;
```

OUTPUT:



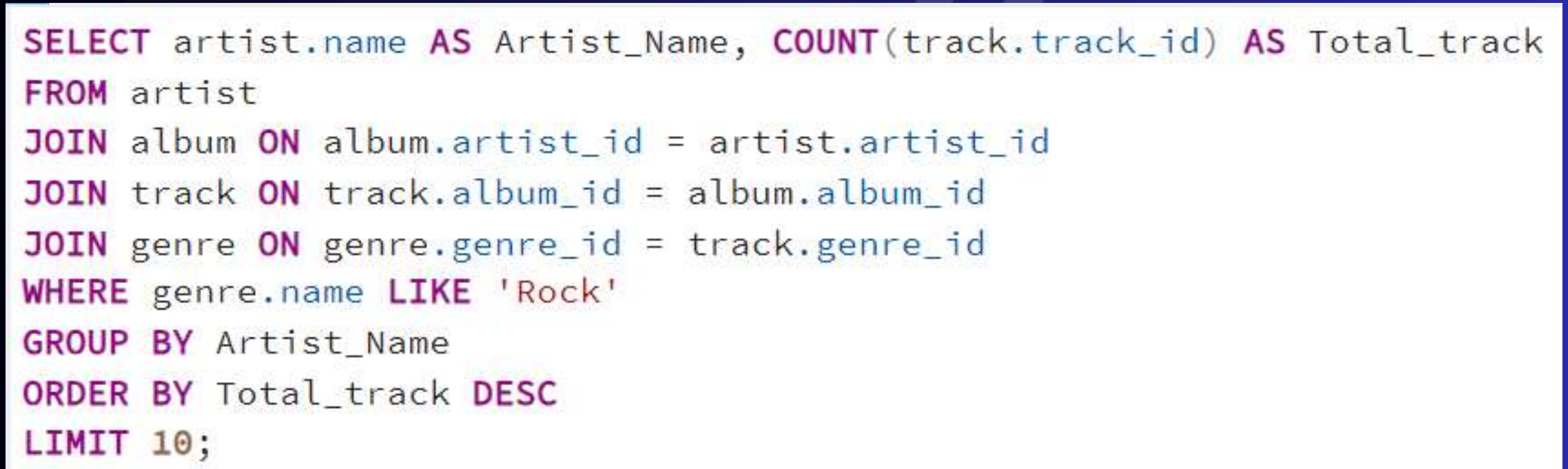
	email character varying (50)	firstname character	lastname character	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron ...	Mitchell ...	Rock
2	alero@uol.com.br	Alexandre ...	Rocha ...	Rock
3	astrid.gruber@apple.at	Astrid	Gruber ...	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen ...	Rock
5	camille.bernard@yahoo.fr	Camille ...	Bernard ...	Rock
6	daan_peeters@apple.be	Daan	Peeters ...	Rock

MODERATE

QUESTION-2

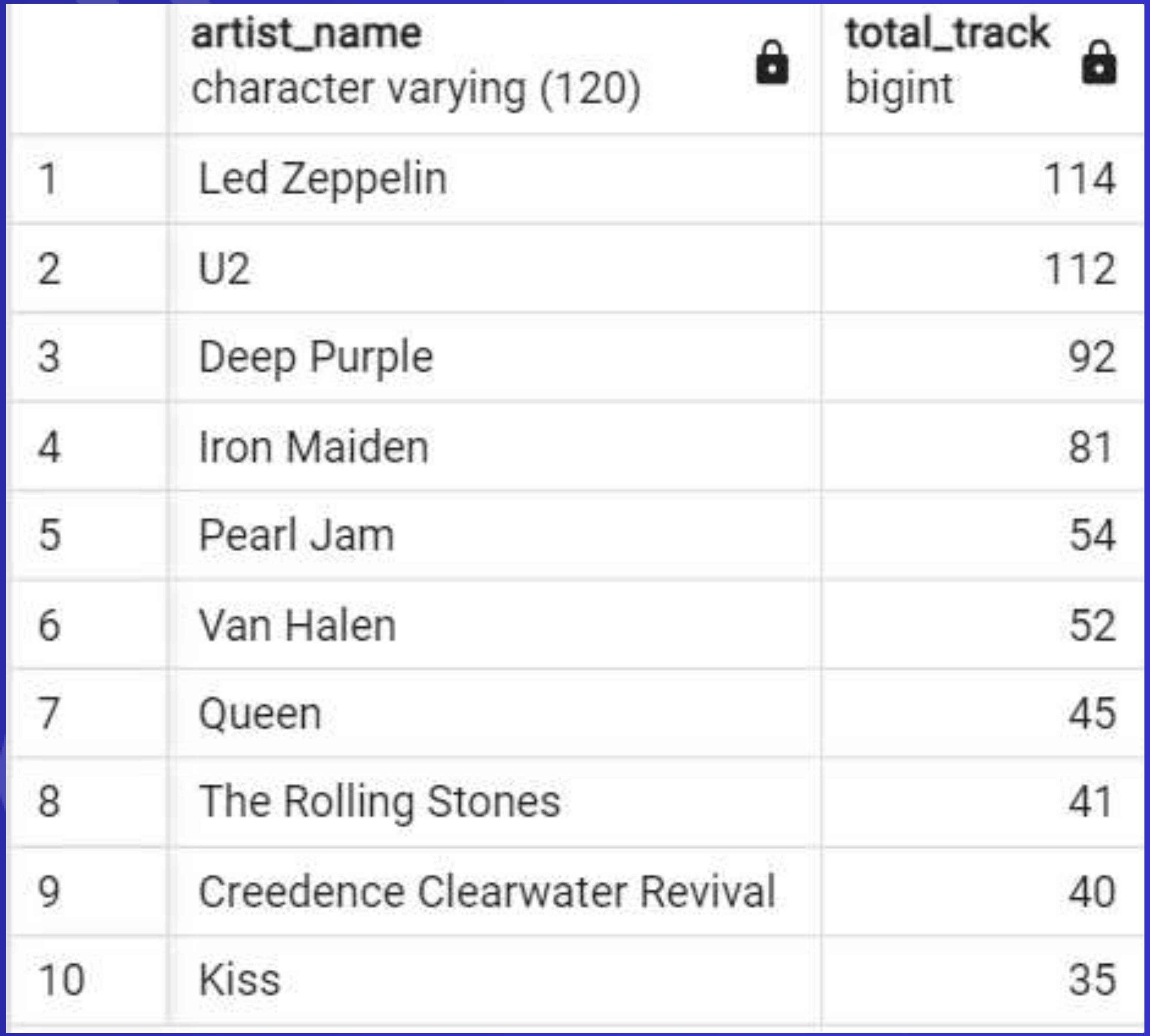
Let's invite the artists who have written the most rock music in our dataset.
Write a query that returns the Artist name and total track count of the top 10 rock bands.

QUERY:



```
SELECT artist.name AS Artist_Name, COUNT(track.track_id) AS Total_track
FROM artist
JOIN album ON album.artist_id = artist.artist_id
JOIN track ON track.album_id = album.album_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY Artist_Name
ORDER BY Total_track DESC
LIMIT 10;
```

OUTPUT:



	artist_name character varying (120)	total_track bigint
1	Led Zeppelin	114
2	U2	112
3	Deep Purple	92
4	Iron Maiden	81
5	Pearl Jam	54
6	Van Halen	52
7	Queen	45
8	The Rolling Stones	41
9	Creedence Clearwater Revival	40
10	Kiss	35

-Pavan Kumar

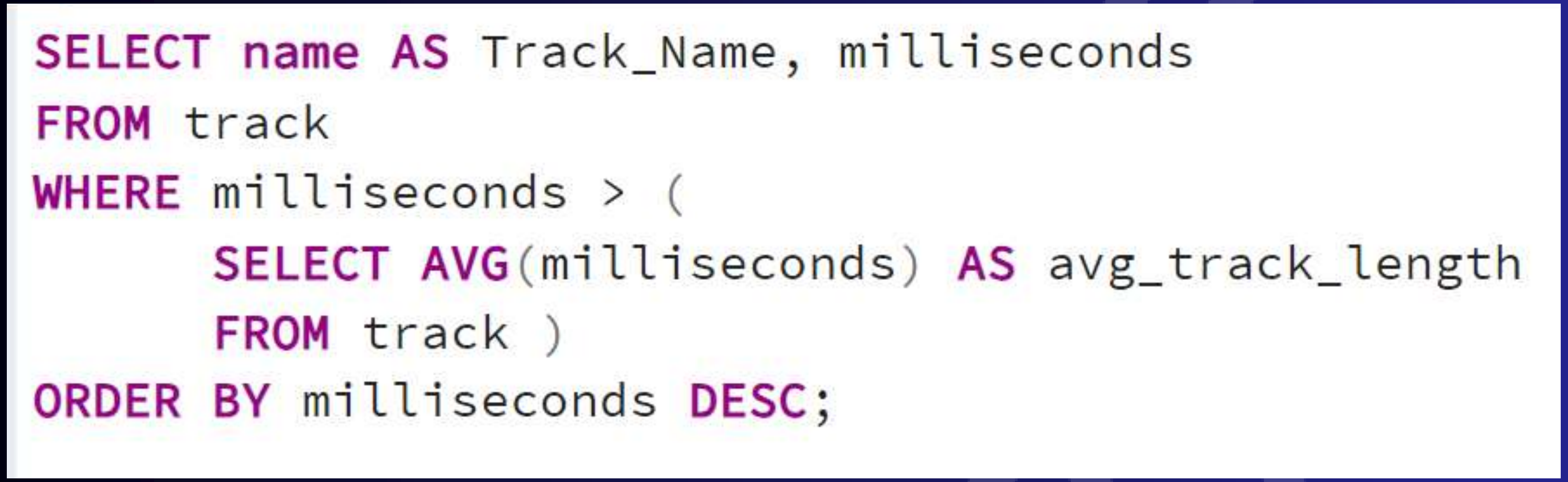
MODERATE

QUESTION-3

Music Store Analysis

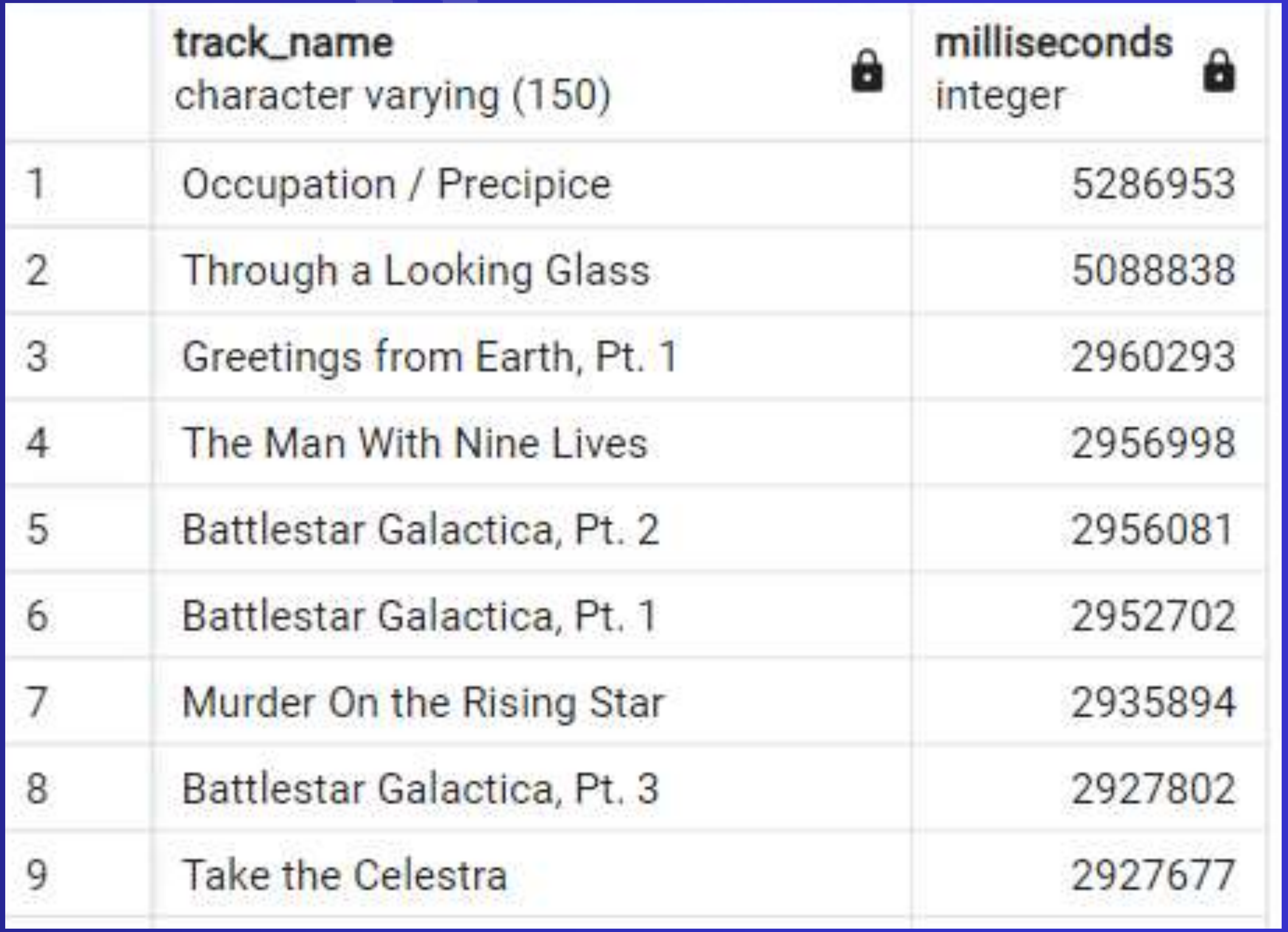
Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

QUERY:



```
SELECT name AS Track_Name, milliseconds
FROM track
WHERE milliseconds > (
      SELECT AVG(milliseconds) AS avg_track_length
      FROM track )
ORDER BY milliseconds DESC;
```

OUTPUT:



	track_name character varying (150) 🔒	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
9	Take the Celestra	2927677

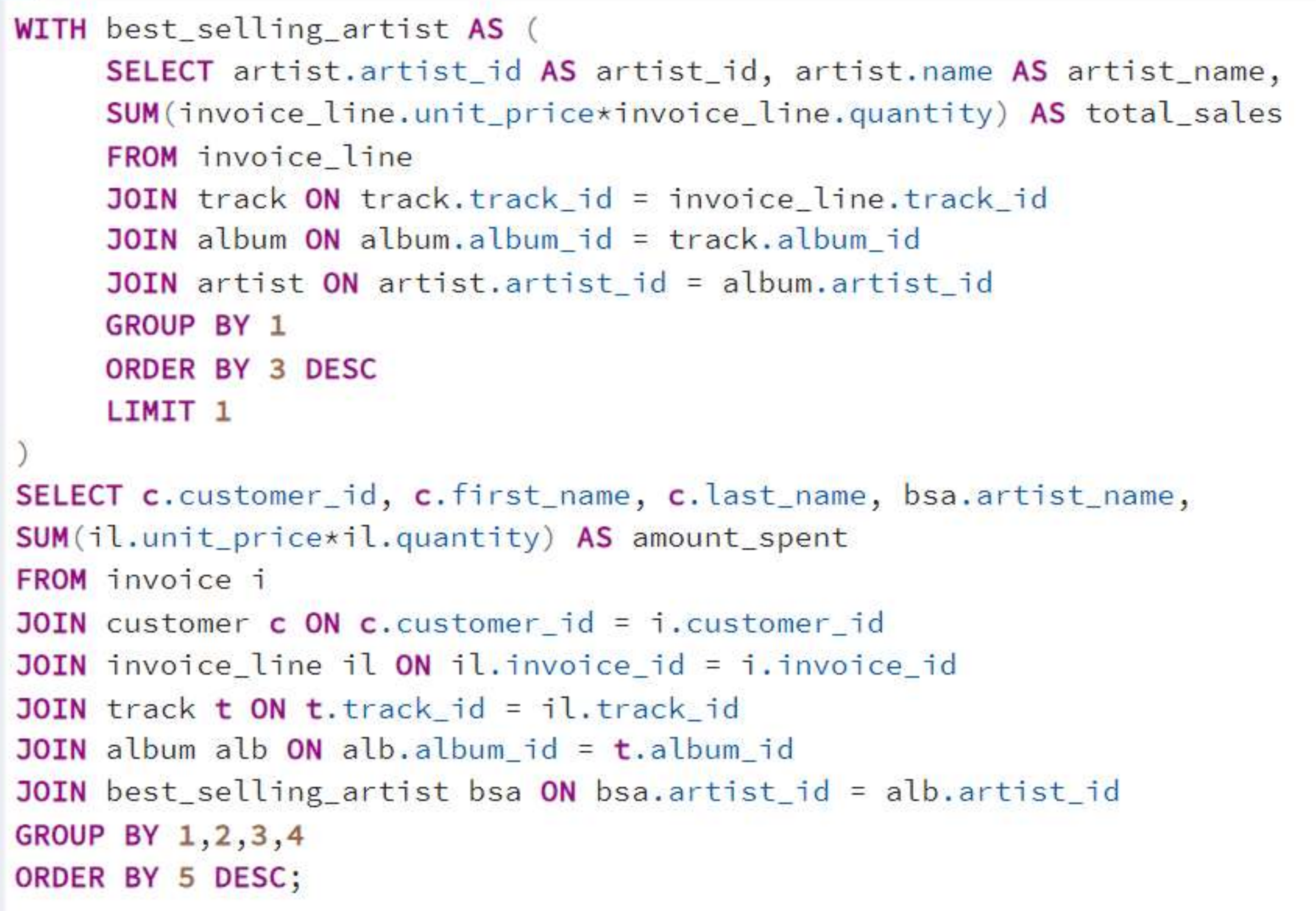
-Pavan Kumar

ADVANCE

QUESTION-1

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

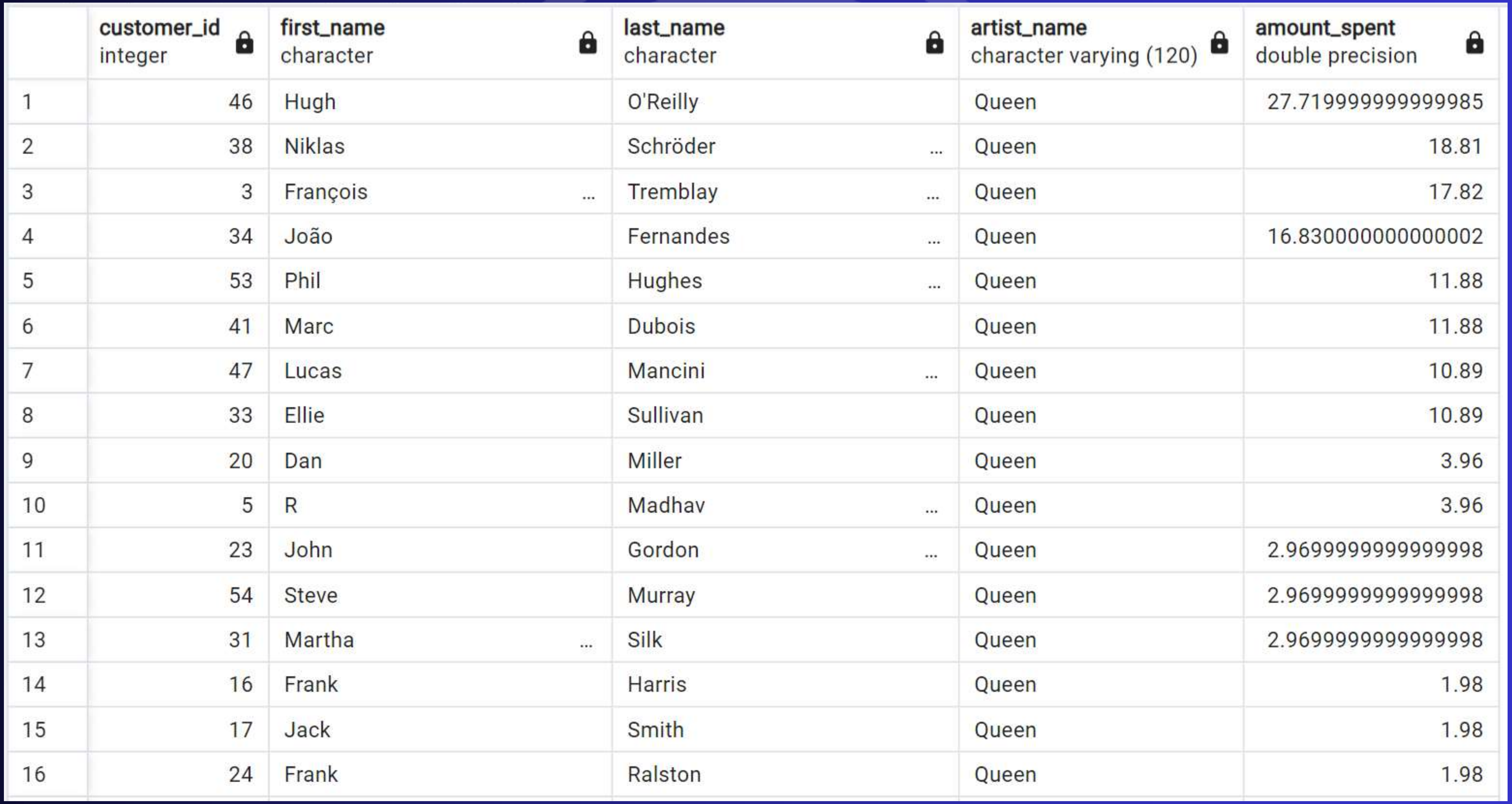
QUERY:



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

ADVANCE

OUTPUT:



	customer_id integer 🔒	first_name character 🔒	last_name character 🔒	artist_name character varying (120) 🔒	amount_spent double precision 🔒
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder ...	Queen	18.81
3	3	François ...	Tremblay ...	Queen	17.82
4	34	João	Fernandes ...	Queen	16.830000000000002
5	53	Phil	Hughes ...	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini ...	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav ...	Queen	3.96
11	23	John	Gordon ...	Queen	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha ...	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98
15	17	Jack	Smith	Queen	1.98
16	24	Frank	Ralston	Queen	1.98

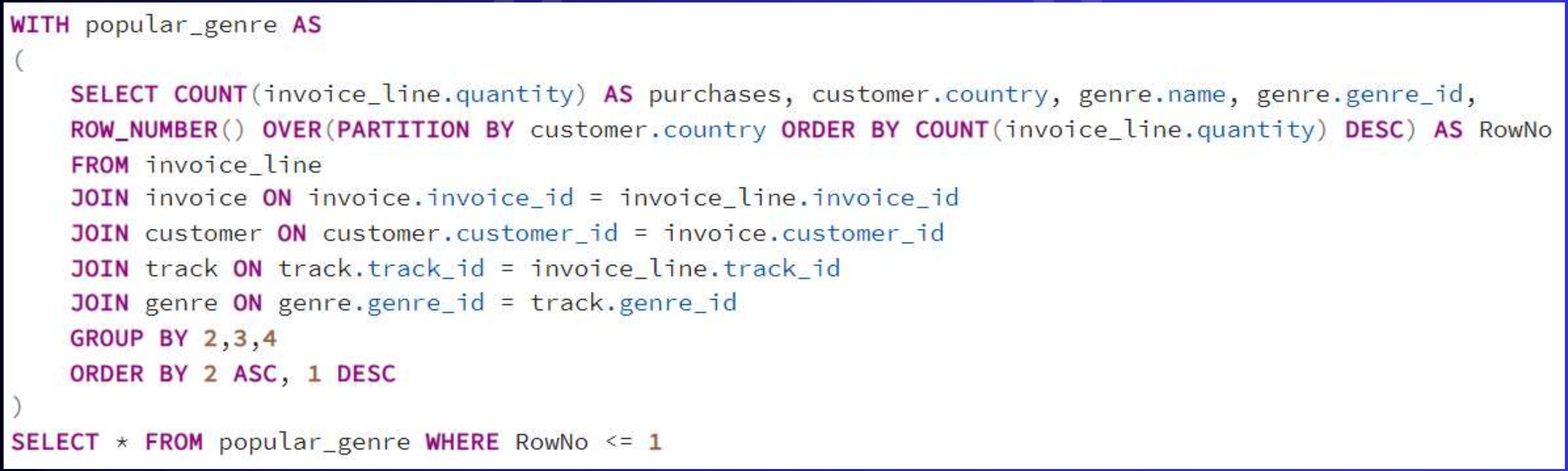
ADVANCE

QUESTION-2

Music Store Analysis

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.

QUERY:

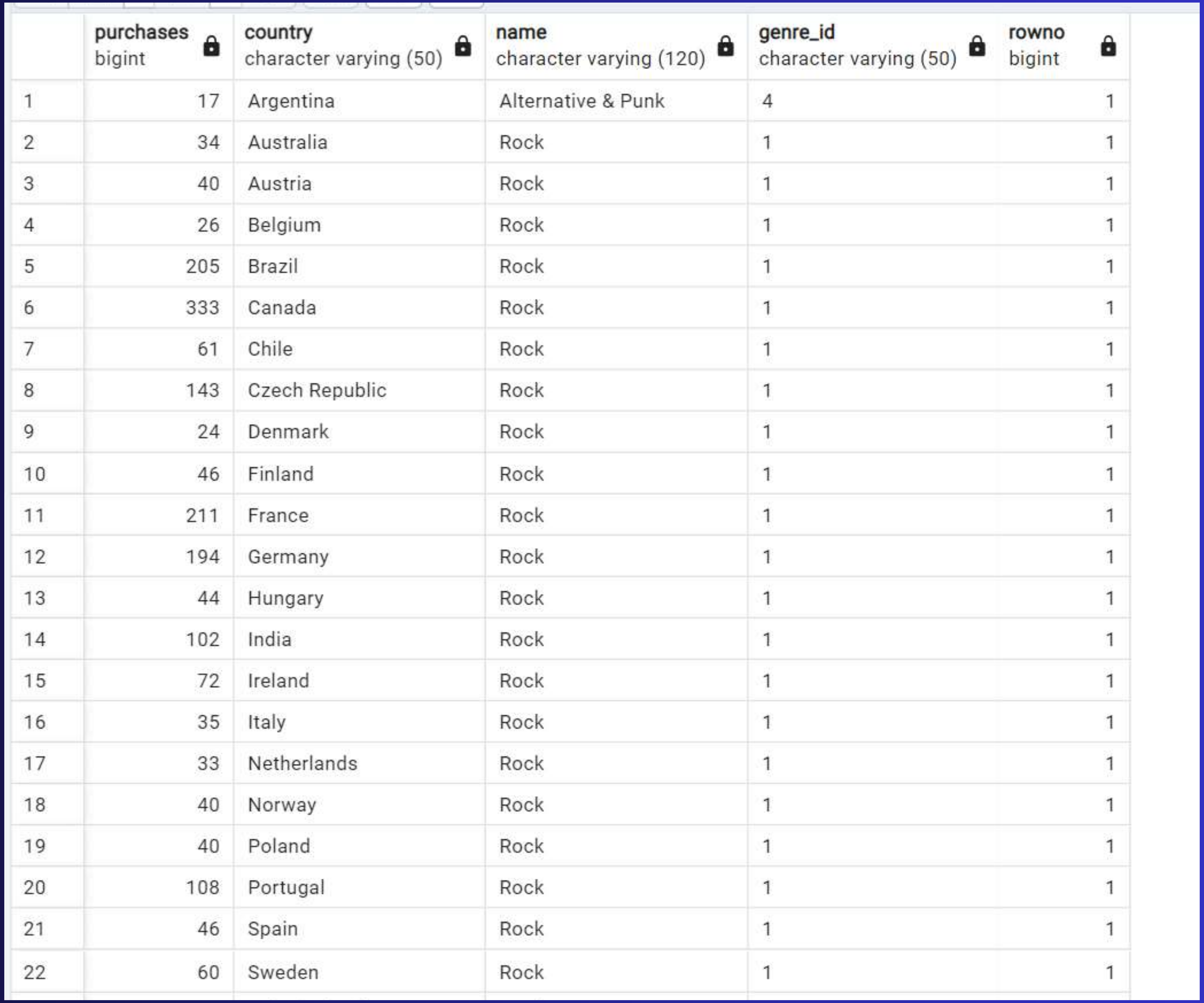


```
WITH popular_genre AS
(
    SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
    ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
    FROM invoice_line
    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
    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
```

-Pavan Kumar

ADVANCE

OUTPUT:



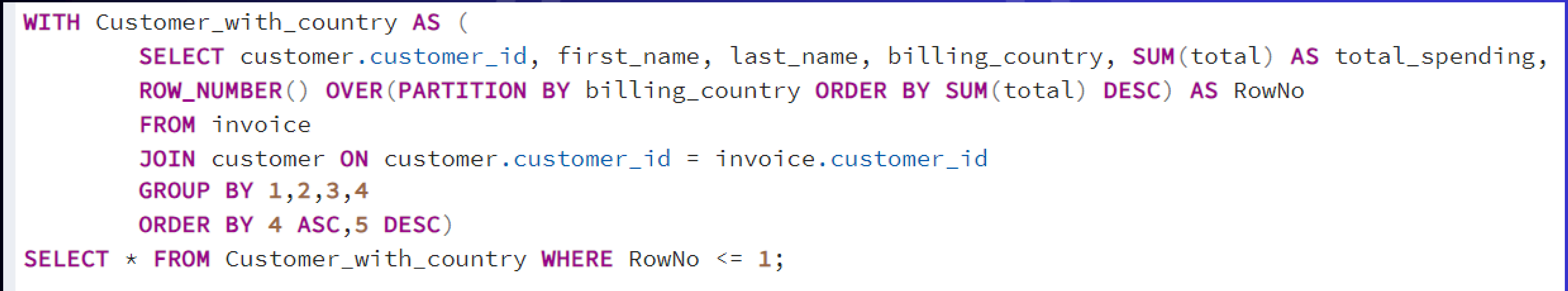
	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1
19	40	Poland	Rock	1	1
20	108	Portugal	Rock	1	1
21	46	Spain	Rock	1	1
22	60	Sweden	Rock	1	1

ADVANCE

QUESTION-3

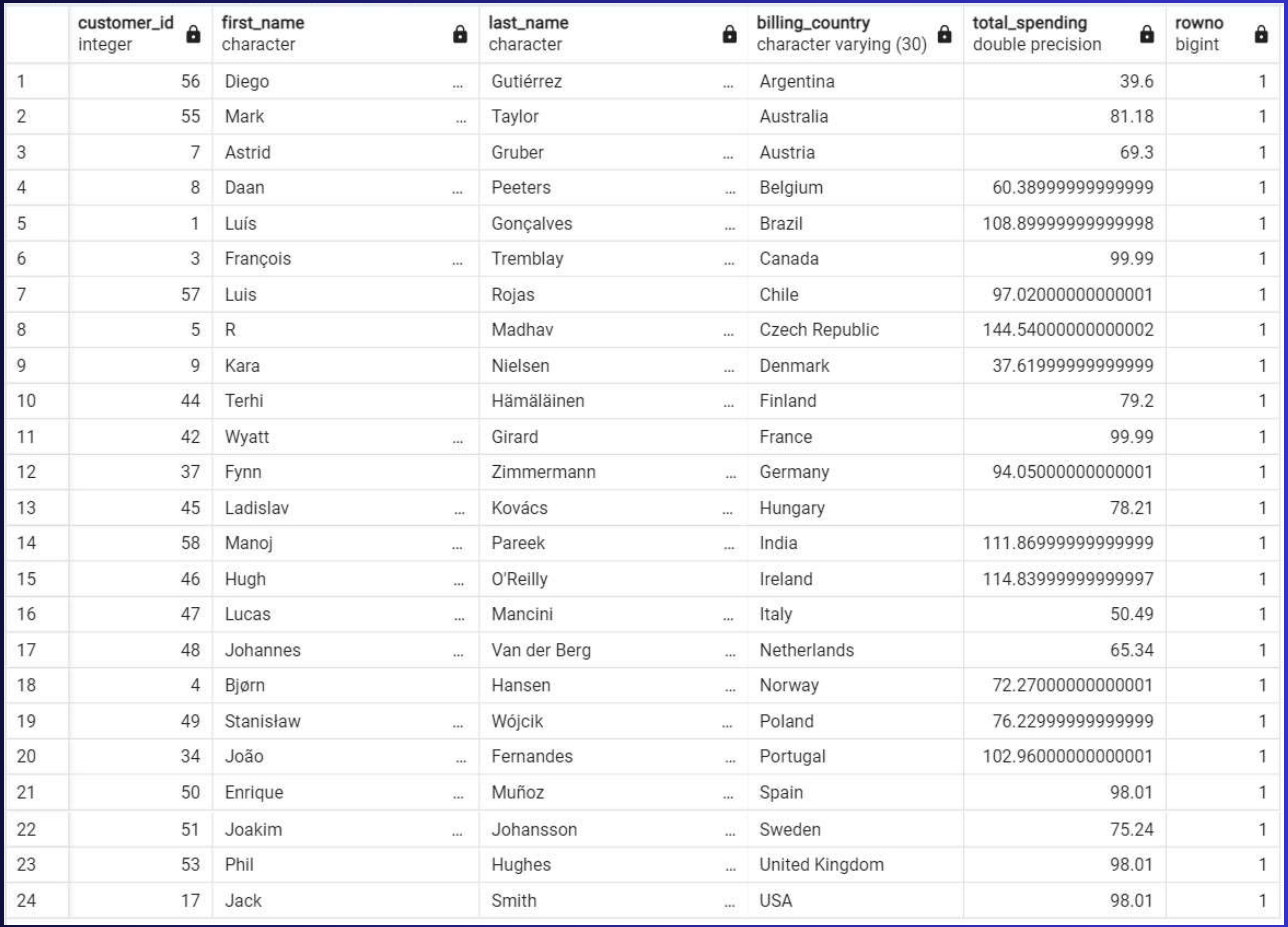
Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.

QUERY:



```
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 invoice
        JOIN 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;
```

OUTPUT:



	customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego	Gutiérrez	Argentina	39.6	1
2	55	Mark	Taylor	Australia	81.18	1
3	7	Astrid	Gruber	Austria	69.3	1
4	8	Daan	Peeters	Belgium	60.38999999999999	1
5	1	Luís	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.05000000000001	1
13	45	Ladislav	Kovács	Hungary	78.21	1
14	58	Manoj	Pareek	India	111.86999999999999	1
15	46	Hugh	O'Reilly	Ireland	114.83999999999997	1
16	47	Lucas	Mancini	Italy	50.49	1
17	48	Johannes	Van der Berg	Netherlands	65.34	1
18	4	Bjørn	Hansen	Norway	72.27000000000001	1
19	49	Stanisław	Wójcik	Poland	76.22999999999999	1
20	34	João	Fernandes	Portugal	102.96000000000001	1
21	50	Enrique	Muñoz	Spain	98.01	1
22	51	Joakim	Johansson	Sweden	75.24	1
23	53	Phil	Hughes	United Kingdom	98.01	1
24	17	Jack	Smith	USA	98.01	1

THANK YOU

in linkedin.com/in/pavankumar12

github.com/pavankumar-11

-Pavan Kumar