SQL PROJECT

DIGITAL MUSIC STORE ANALYSIS

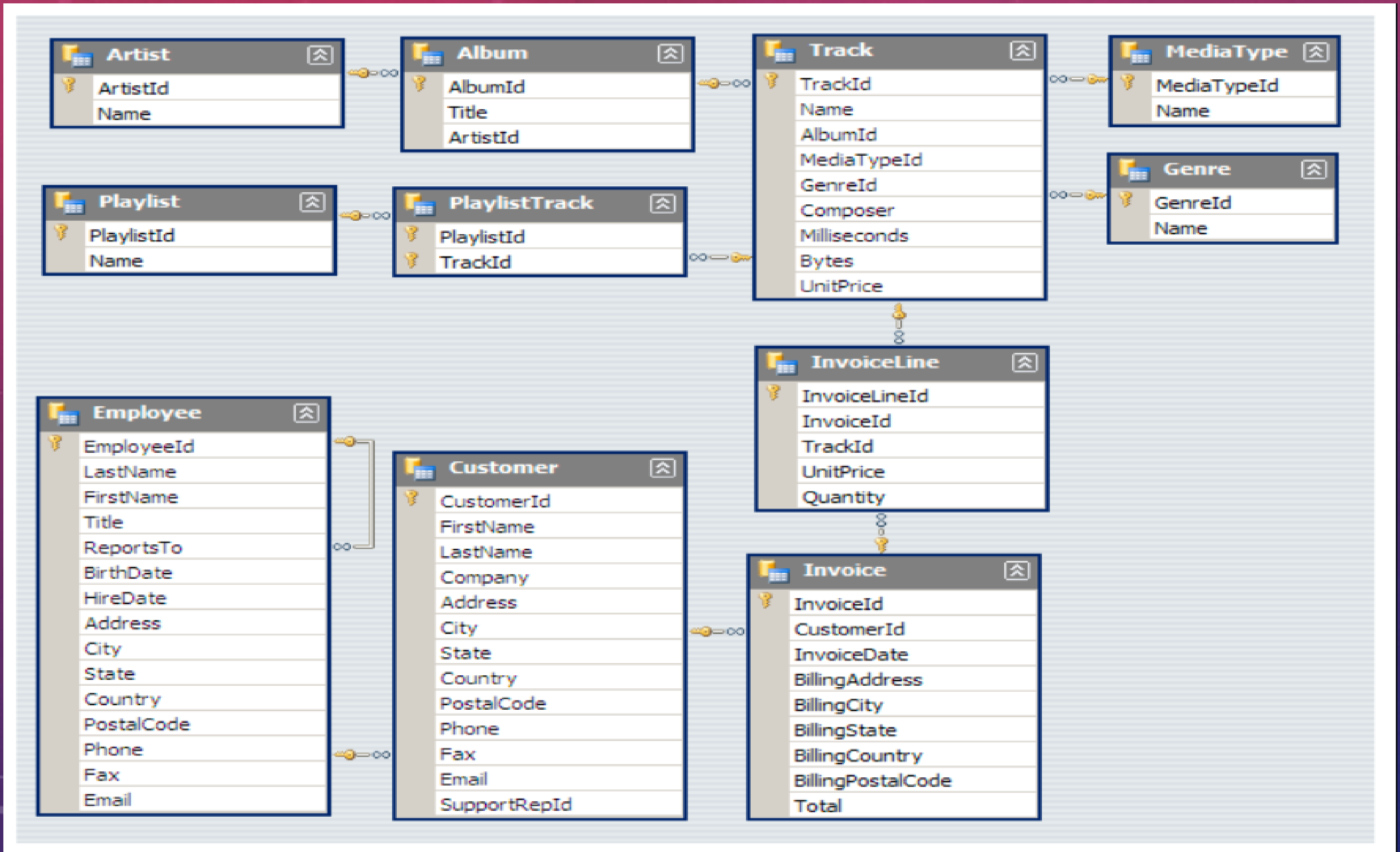
BY KESHAV SHARMA

OBJECTIVE

- ❑ *THIS PROJECT AIMS TO ANALYSE A DIGITAL MUSIC STORE DATABASE USING SQL PROVIDING STAKEHOLDERS WITH VALUABLE INSIGHTS FOR DECISION MAKING.*

- ❑ *THROUGH SQL QUERIES, IT ADDRESSES QUESTIONS GENRE PERFORMANCE,REVENUE ETC.WHICH CAN HELP THE MUSIC STORE UNDERSTAND ITS GROWTH BY ANSWERING SIMPLE QUESTIONS.*

DATABASE SCHEMA



Artist
- ArtistId
- Name

Album
- AlbumId
- Title
- ArtistId

Track
- TrackId
- Name
- AlbumId
- MediaTypeId
- GenreId
- Composer
- Milliseconds
- Bytes
- UnitPrice

MediaType
- MediaTypeId
- Name

Genre
- GenreId
- Name

Playlist
- PlaylistId
- Name

PlaylistTrack
- PlaylistId
- TrackId

InvoiceLine
- InvoiceLineId
- InvoiceId
- TrackId
- UnitPrice
- Quantity

Employee
- EmployeeId
- LastName
- FirstName
- Title
- ReportsTo
- BirthDate
- HireDate
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email

Customer
- CustomerId
- FirstName
- LastName
- Company
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email
- SupportRepId

Invoice
- InvoiceId
- CustomerId
- InvoiceDate
- BillingAddress
- BillingCity
- BillingState
- BillingCountry
- BillingPostalCode
- Total

LEVEL OF QUESTIONS

EASY

MODERATE

ADVANCE

QUERIES INCLUDES
SELECT,GROUP BY
.ORDER BY,LIMIT

QUERIES INCLUDES
JOINS,SUB-QUERIES

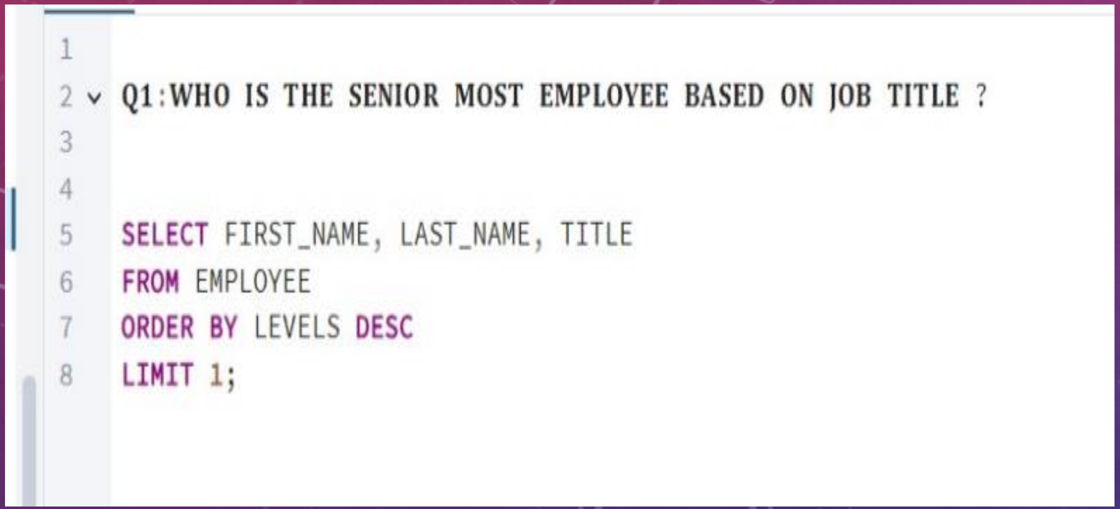
QUERIES INCLUDES
CTE (COMMON TABLE
EXPRESSION)

QUESTION 1

EASY

WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE?

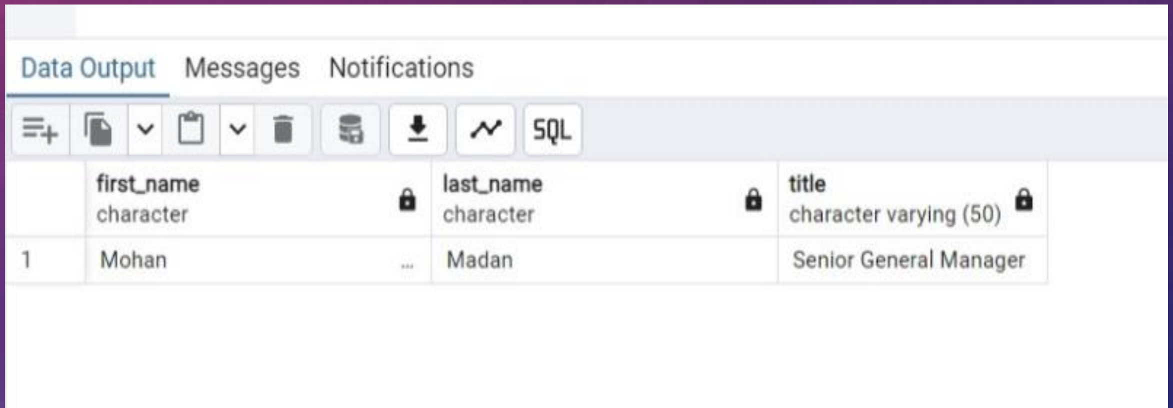
QUERY



```
Q1:WHO IS THE SENIOR MOST EMPLOYEE BASED ON JOB TITLE ?


SELECT FIRST_NAME, LAST_NAME, TITLE
FROM EMPLOYEE
ORDER BY LEVELS DESC
LIMIT 1;
```

OUTPUT

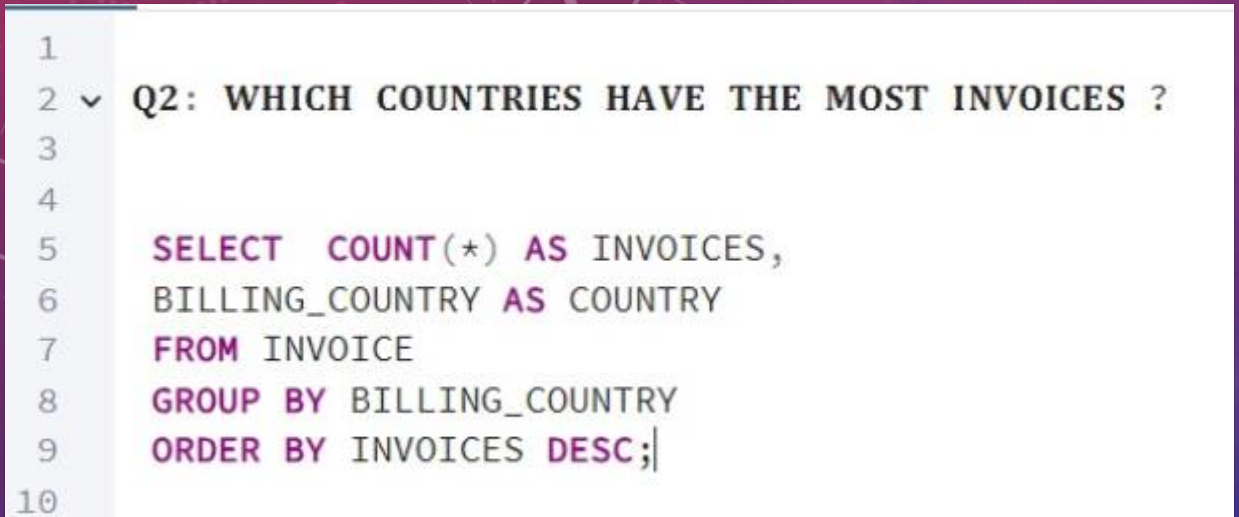


	first_name character	last_name character	title character varying (50)
1	Mohan	Madan	Senior General Manager

QUESTION 2 EASY

WHICH COUNTRIES HAVE THE MOST INVOICES?

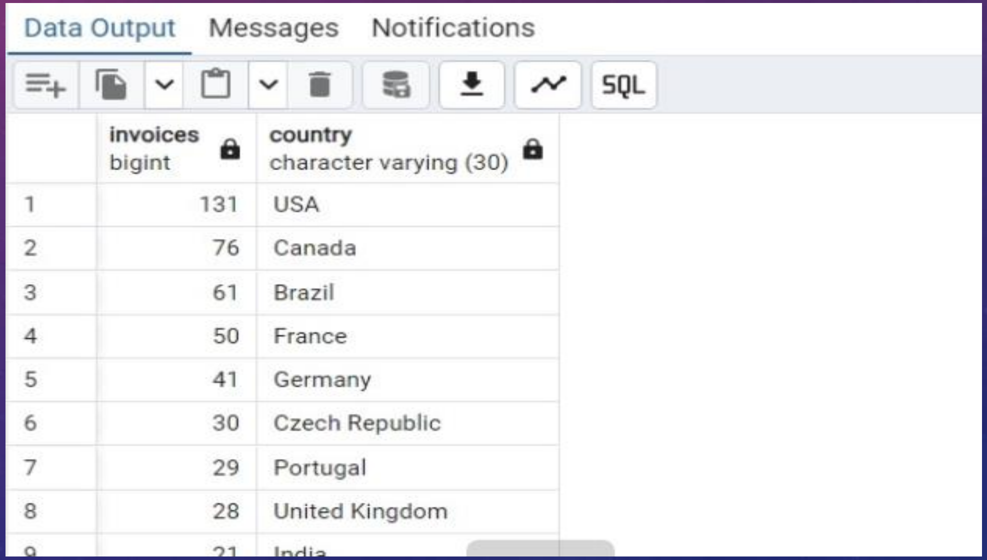
QUERY



```
Q2: WHICH COUNTRIES HAVE THE MOST INVOICES ?


 SELECT  COUNT(*) AS INVOICES,
 BILLING_COUNTRY AS COUNTRY
 FROM INVOICE
 GROUP BY BILLING_COUNTRY
 ORDER BY INVOICES DESC;
```

OUTPUT



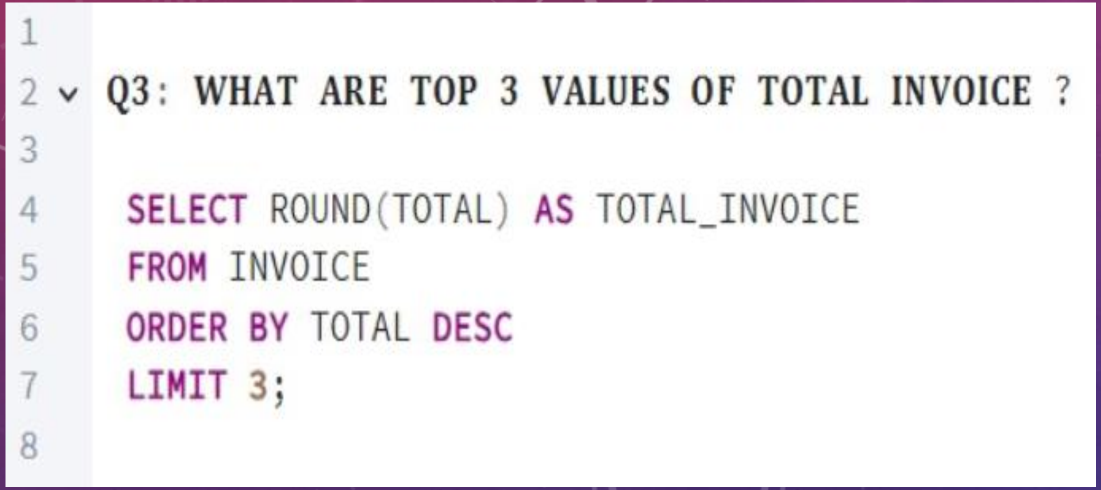
```
     invoices   country
     bigint     character varying (30)
1         131   USA
2          76   Canada
3          61   Brazil
4          50   France
5          41   Germany
6          30   Czech Republic
7          29   Portugal
8          28   United Kingdom
9          21   India
```

QUESTION 3

EASY

WHAT ARE THE TOP 3 VALUES OF TOTAL INVOICE?

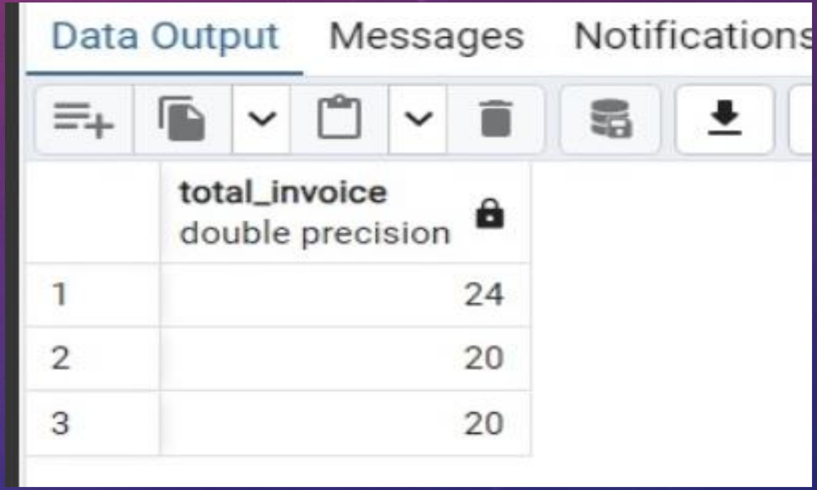
QUERY



```
Q3: WHAT ARE TOP 3 VALUES OF TOTAL INVOICE ?

 SELECT ROUND(TOTAL) AS TOTAL_INVOICE
 FROM INVOICE
 ORDER BY TOTAL DESC
 LIMIT 3;
```

OUTPUT

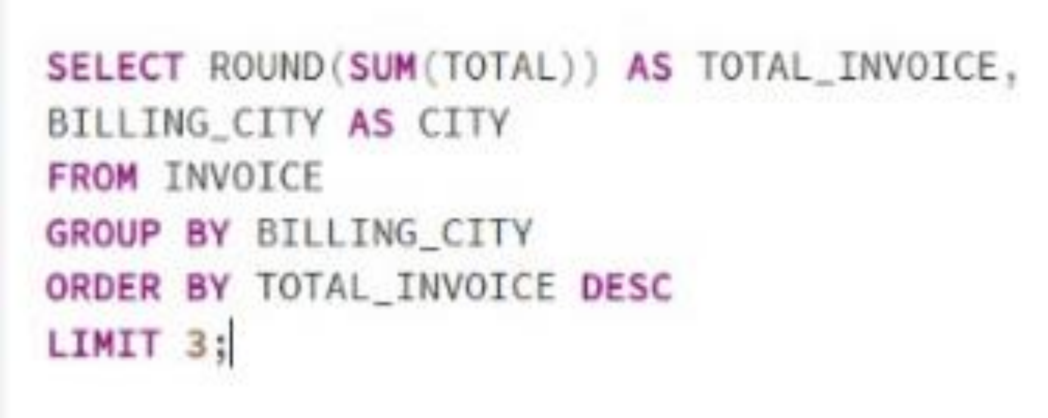


	total_invoice double precision
1	24
2	20
3	20

QUESTION 4 EASY

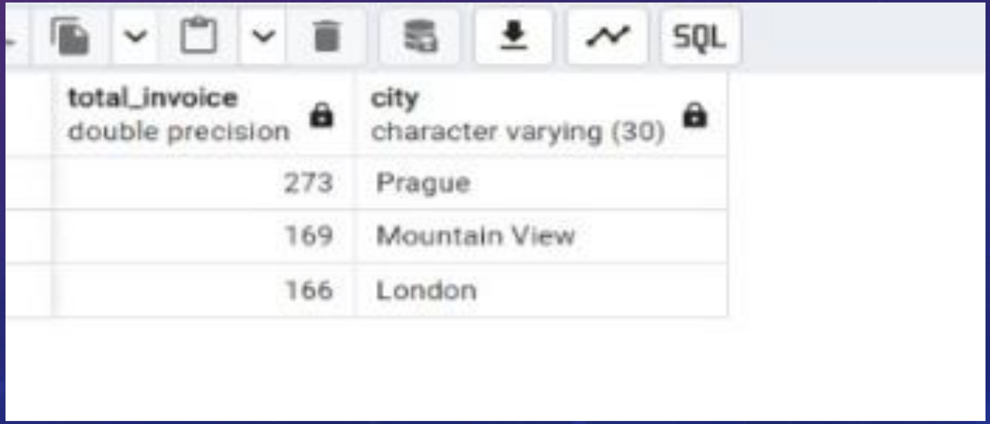
WHICH CITY HAS THE BEST CUSTOMERS? WE WOULD LIKE TO THROW A PROMOTIONAL MUSIC FESTIVAL IN THE CITY WE MADE THE MOST MONEY. WRITE A QUERY THAT RETURNS ONE CITY THAT HAS THE HIGHEST SUM OF INVOICE TOTALS.RETURN BOTH THE CITY NAME AND SUM OF ALL INVOICE TOTALS.

QUERY



```
SELECT ROUND(SUM(TOTAL)) AS TOTAL_INVOICE,
BILLING_CITY AS CITY
FROM INVOICE
GROUP BY BILLING_CITY
ORDER BY TOTAL_INVOICE DESC
LIMIT 3;
```

OUTPUT



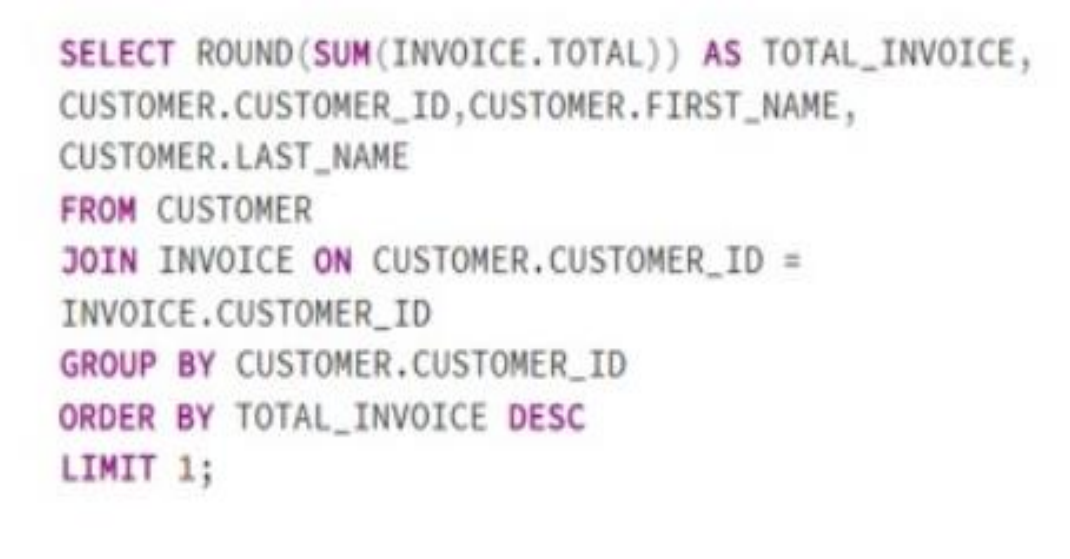
total_invoice double precision	city character varying (30)
273	Prague
169	Mountain View
166	London

QUESTION 5

EASY

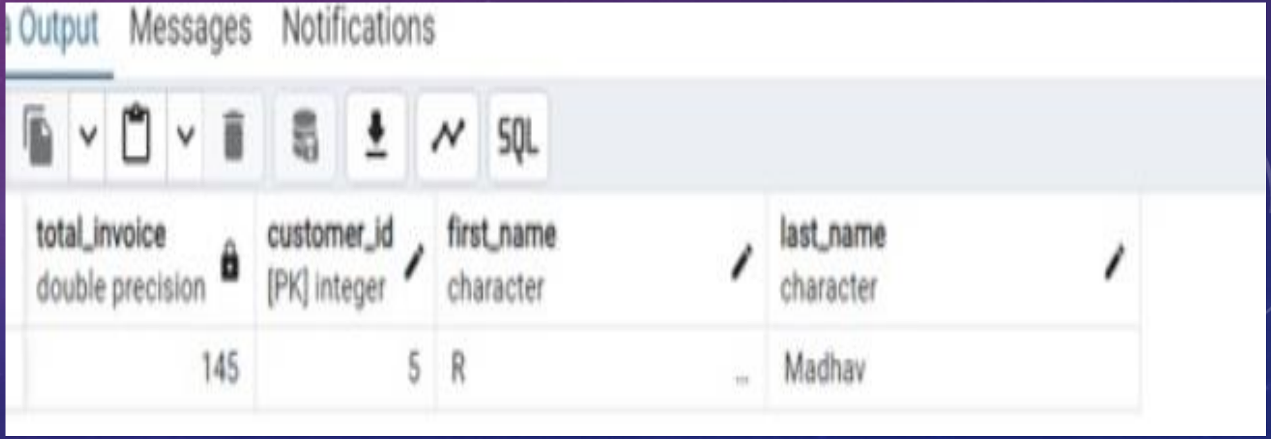
WHO IS THE BEST CUSTOMER? THE CUSTOMER WHO HAS SPENT THE MOST MONEY WILL BE DECLARED THE BEST CUSTOMER. WRITE A QUERY THAT RETURNS PERSON WHO HAS SPEND THE MOST MONEY.

QUERY



```sql
SELECT ROUND(SUM(INVOICE.TOTAL)) AS TOTAL_INVOICE,
CUSTOMER.CUSTOMER_ID,CUSTOMER.FIRST_NAME,
CUSTOMER.LAST_NAME
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CUSTOMER_ID =
INVOICE.CUSTOMER_ID
GROUP BY CUSTOMER.CUSTOMER_ID
ORDER BY TOTAL_INVOICE DESC
LIMIT 1;
```

OUTPUT

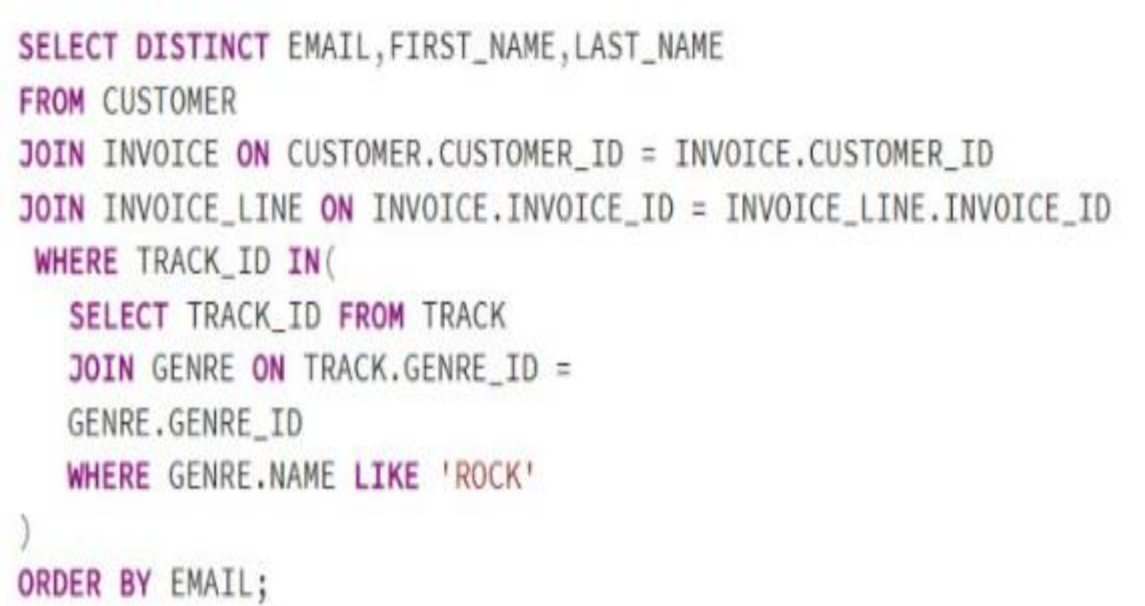


total_invoice double precision	customer_id [PK] integer	first_name character	last_name character
145	5	R	Madhav

QUESTION 1 MODERATE

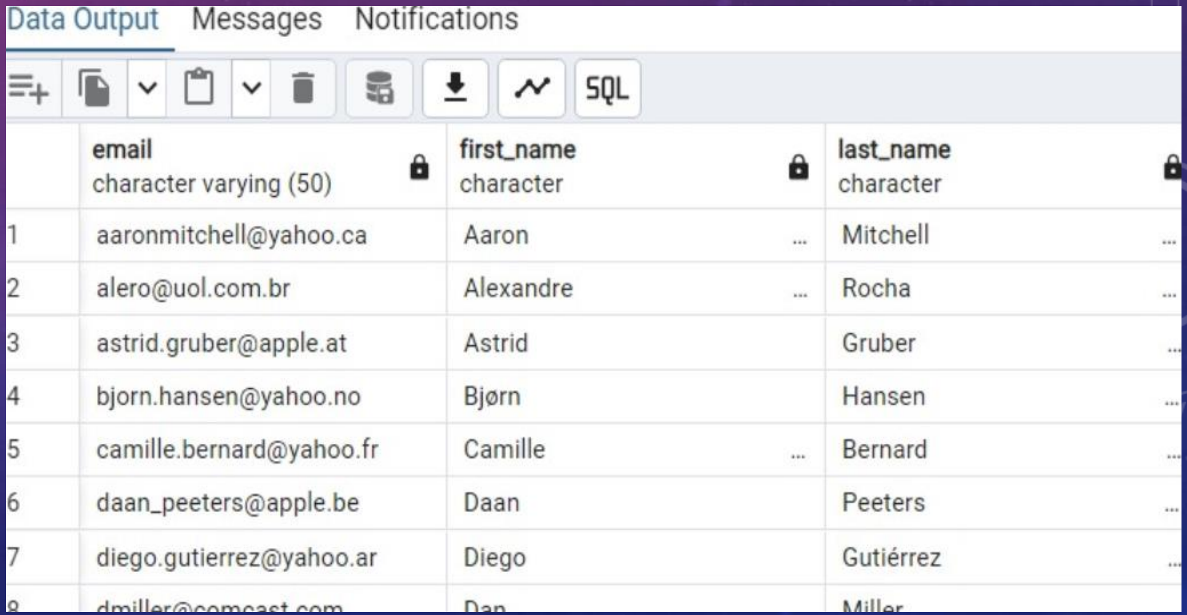
WRITE A QUERY TO RETURN THE EMAIL,FIRST NAME,LAST NAME, AND GENRE OF ALL ROCK MUSIC LISTENERS. RETURN YOUR LIST ORDERED ALPHABETICALLY BY EMAIL STARTING WITH A.

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

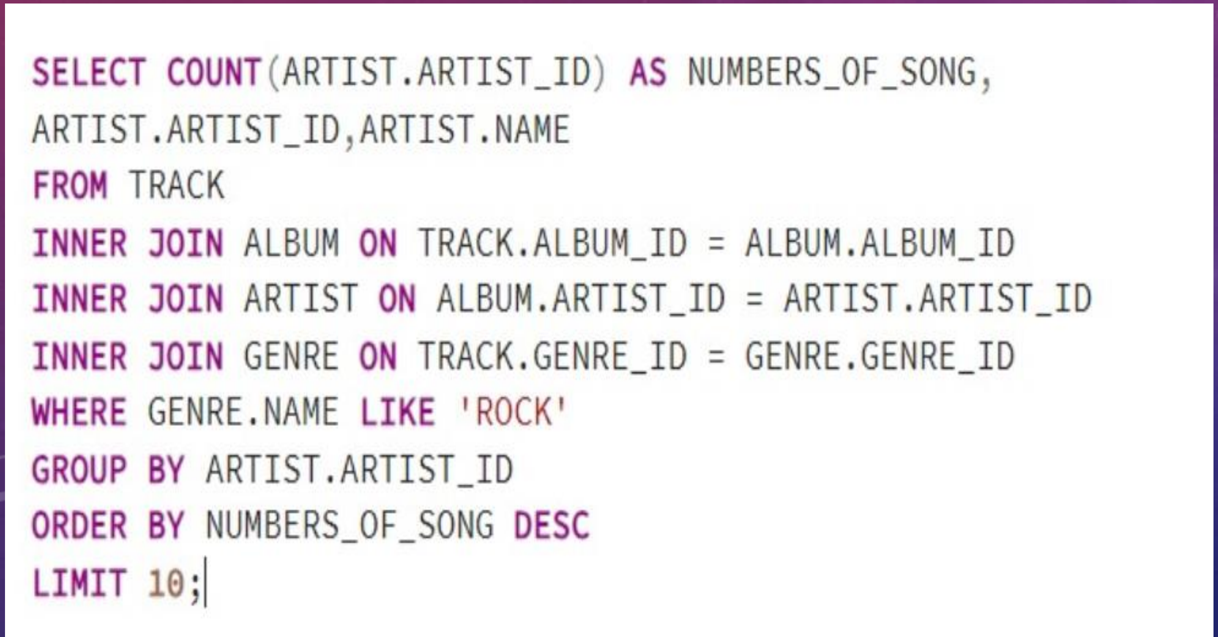


	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller

QUESTION 2 MODERATE

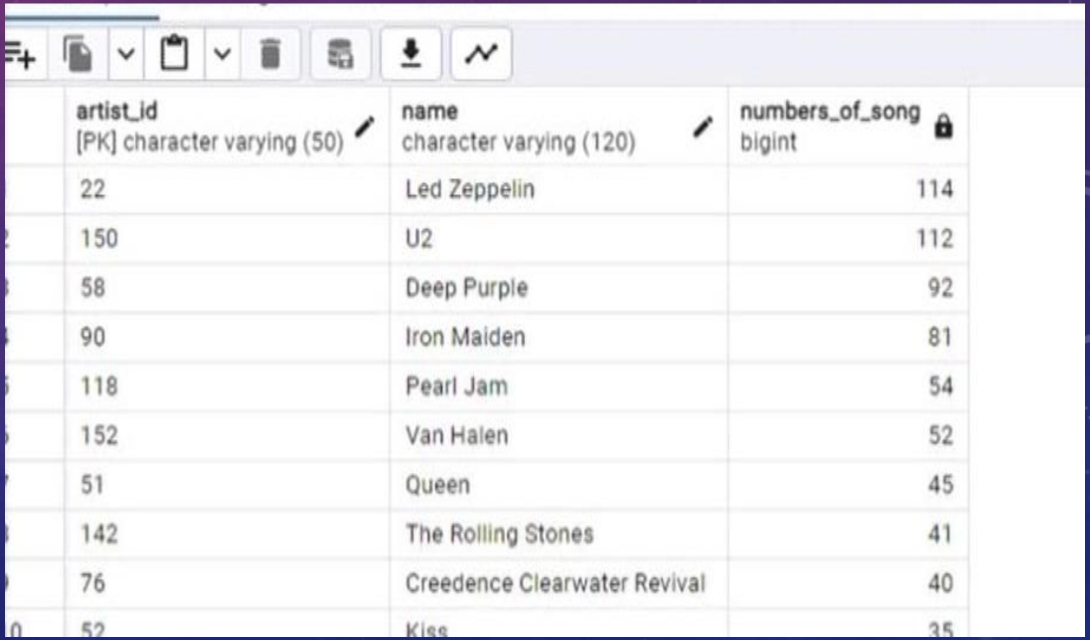
LETS INVITE THE ARTISTS WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR DATASET. WRITE A QUERY THAT RETURNS THE ARTIST NAME AND TOTAL TRACK COUNT OF THE TOP 10 ROCK BANDS.

QUERY



```
SELECT COUNT(ARTIST.ARTIST_ID) AS NUMBERS_OF_SONG,
ARTIST.ARTIST_ID,ARTIST.NAME
FROM TRACK
INNER JOIN ALBUM ON TRACK.ALBUM_ID = ALBUM.ALBUM_ID
INNER JOIN ARTIST ON ALBUM.ARTIST_ID = ARTIST.ARTIST_ID
INNER JOIN GENRE ON TRACK.GENRE_ID = GENRE.GENRE_ID
WHERE GENRE.NAME LIKE 'ROCK'
GROUP BY ARTIST.ARTIST_ID
ORDER BY NUMBERS_OF_SONG DESC
LIMIT 10;
```

OUTPUT

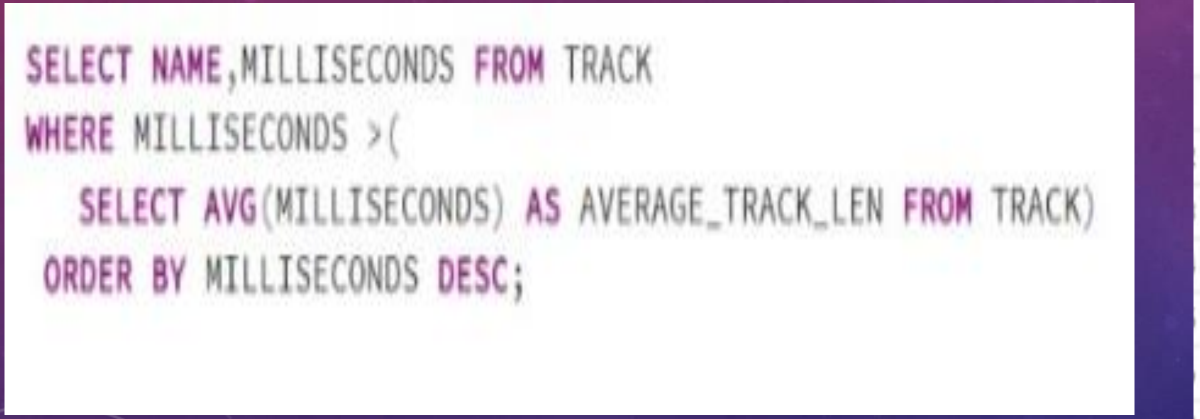


artist_id [PK] character varying (50)	name character varying (120)	numbers_of_song bigint
22	Led Zeppelin	114
150	U2	112
58	Deep Purple	92
90	Iron Maiden	81
118	Pearl Jam	54
152	Van Halen	52
51	Queen	45
142	The Rolling Stones	41
76	Creedence Clearwater Revival	40
52	Kiss	35

QUESTION 3 MODERATE

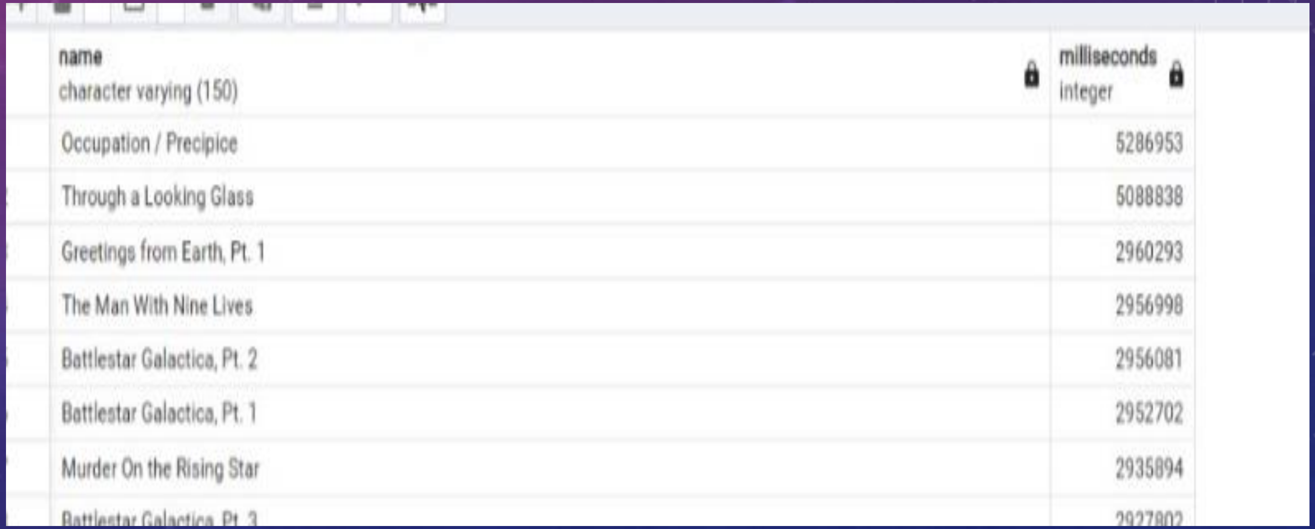
RETURN ALL THE TRACK NAMES THAT HAVE A SONG LENGTH LONGER THAN THE AVERAGE SONG LENGTH. RETURN THE NAME AND MILLISECONDS FOR EACH TRACK. ORDER BY SONG LENGTH WITH THE LONGEST SONG LISTED FIRST.

QUERY



```
SELECT NAME,MILLISECONDS FROM TRACK
WHERE MILLISECONDS >(
   SELECT AVG(MILLISECONDS) AS AVERAGE_TRACK_LEN FROM TRACK)
 ORDER BY MILLISECONDS DESC;
```

OUTPUT

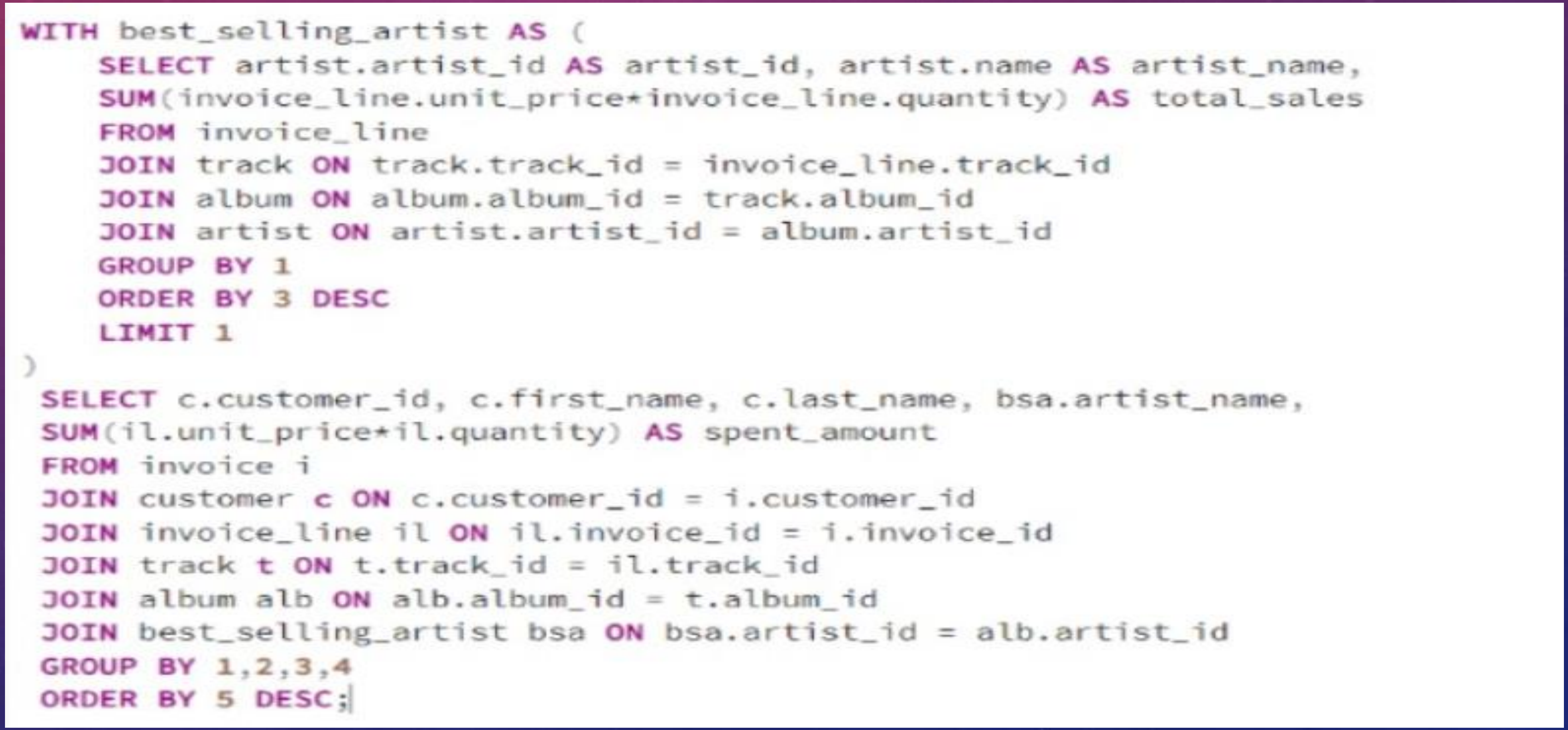


name character varying (150)	milliseconds integer
Occupation / Precipice	5286953
Through a Looking Glass	5088838
Greetings from Earth, Pt. 1	2960293
The Man With Nine Lives	2956998
Battlestar Galactica, Pt. 2	2956081
Battlestar Galactica, Pt. 1	2952702
Murder On the Rising Star	2935894
Battlestar Galactica, Pt. 3	2927802

QUESTION 1 ADVANCE

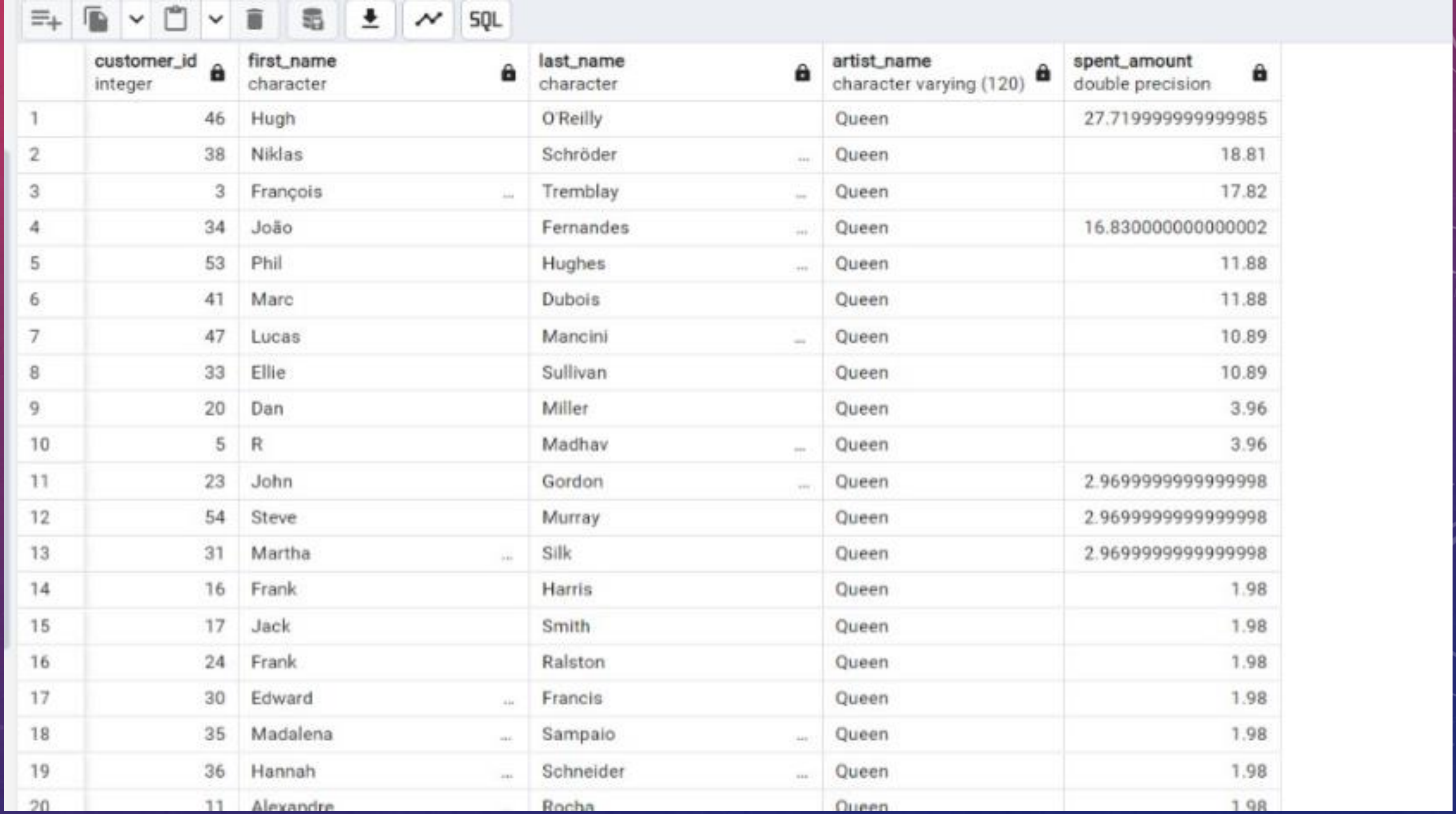
FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME , ARTIST NAME , AND TOTAL SPENT

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 spent_amount
 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;
```

OUTPUT :

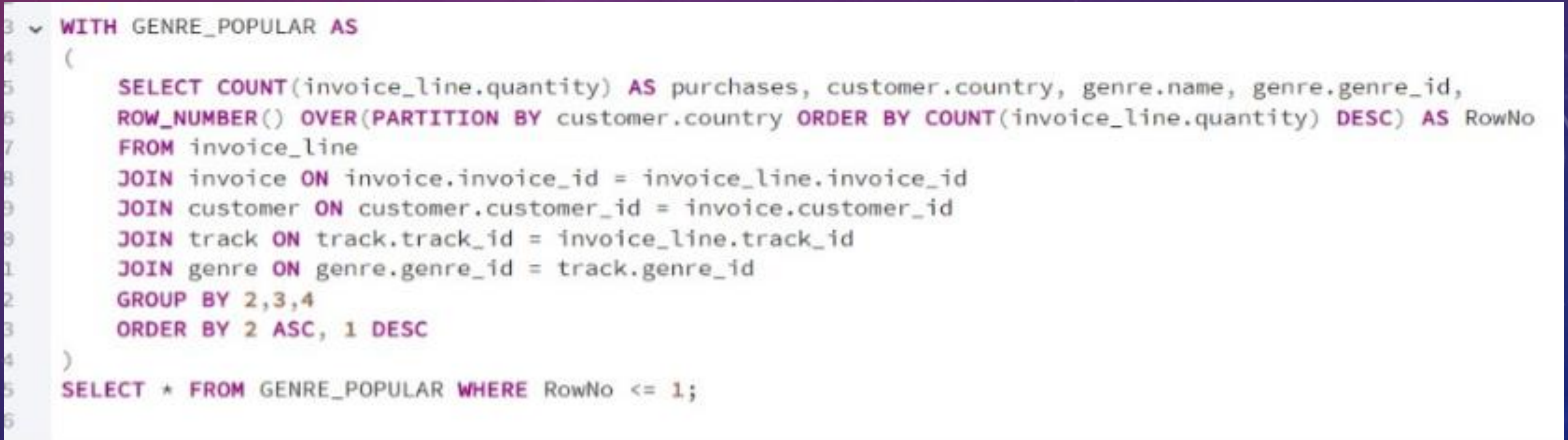


	customer_id integer	first_name character	last_name character	artist_name character varying (120)	spent_amount double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	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
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98
19	36	Hannah	Schneider	Queen	1.98
20	11	Alexandre	Rocha	Queen	1.98

QUESTION 2 ADVANCE

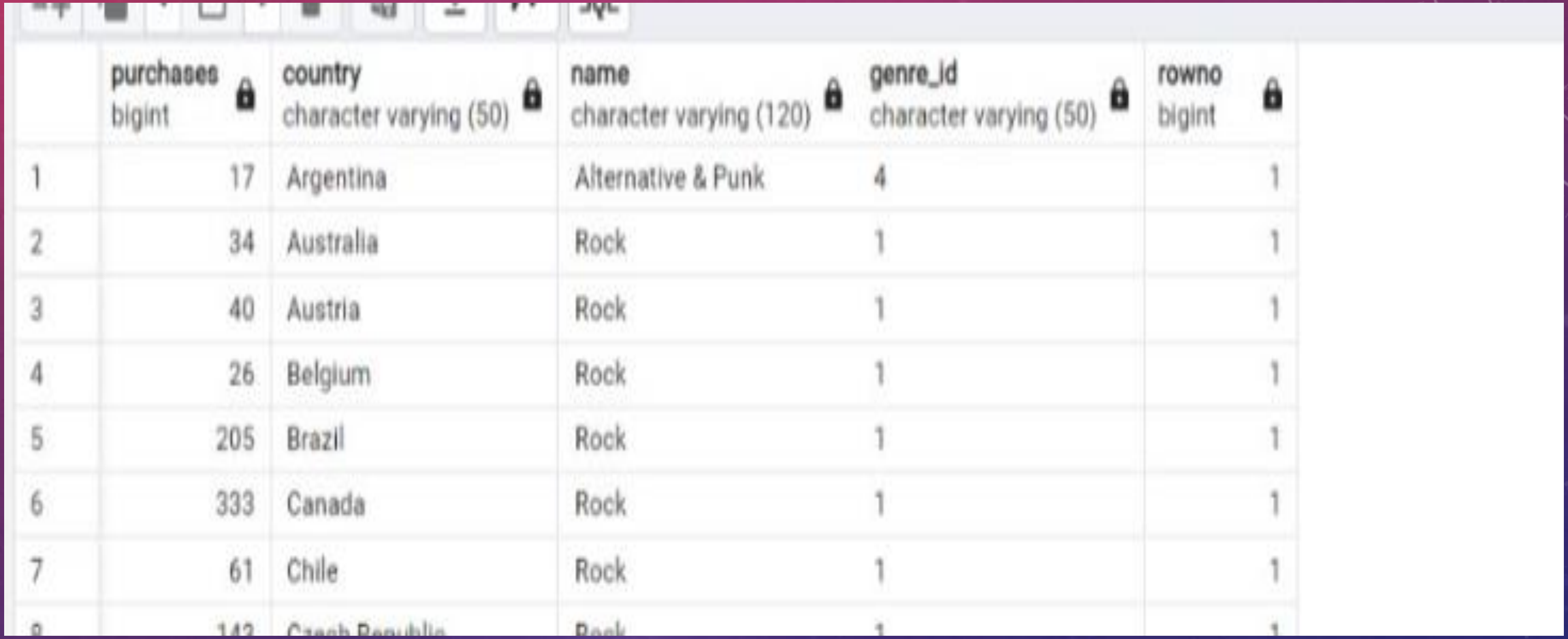
WE WANT TO FIND OUT THE MOST POPULAR MUSIC GENRE FOR EACH COUNTRY.WE DETERMINE THE MOST POPULAR GENRE AS THE GENRE WITH THE HIGHEST AMOUNT OF PURCHASES. WRITE A QUERY THAT RETURNS EACH COUNTRY ALONG WITH THE TOP GENRE. FOR COUNTRIES WHERE THE MAXIMUM NUMBER OF PURCHASES IS SHARED RETURN ALL GENRES.

- QUERY :



```
WITH GENRE_POPULAR 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 GENRE_POPULAR WHERE RowNo <= 1;
```

OUTPUT :

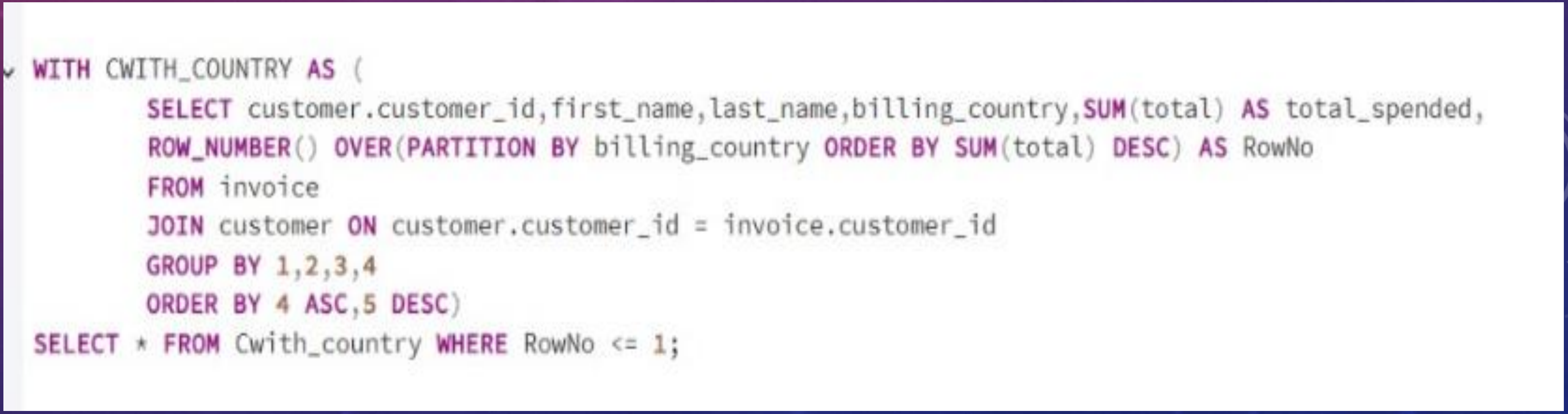


	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1

QUESTION 3 ADVANCE

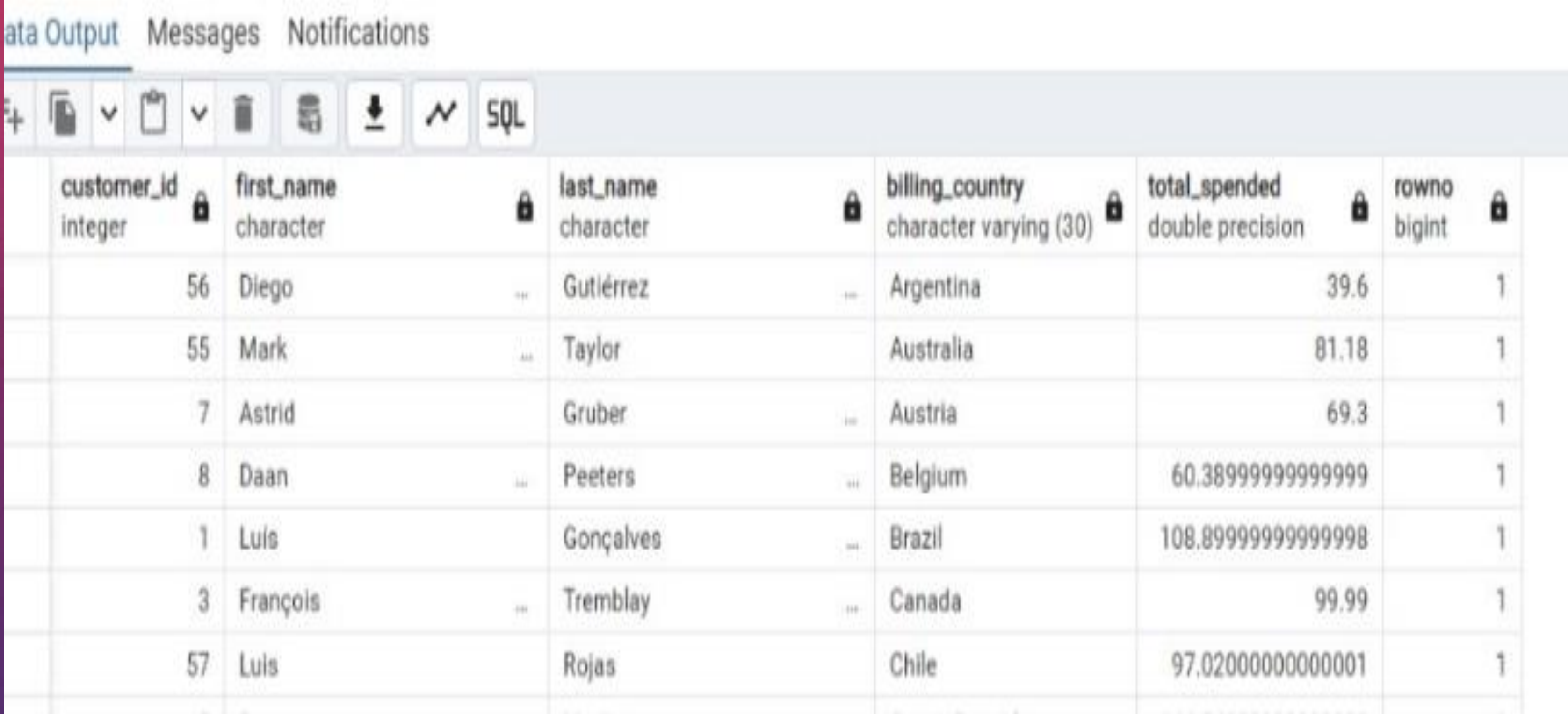
WRITE A QUERY THAT DETERMINES THE CUSTOMER THAT HAS SPEND THE MOST ON MUSIC FOR EACH COUNTRY. WRITE A QUERY THAT RETURNS THE COUNTRY ALONG WITH THE TOP CUSTOMER AND HOW MUCH THEY SPENT. FOR COUNTRIES WHERE THE TOP AMOUNT SPENT IS SHARED, PROVIDE ALL CUSTOMERS WHO SPENT THIS AMOUNT.

QUERY :



```sql
WITH CWITH_COUNTRY AS (
        SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spended,
        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 Cwith_country WHERE RowNo <= 1;
```

OUTPUT :



customer_id integer	first_name character	last_name character	billing_country character varying (30)	total_spended double precision	rowno bigint
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

THANK YOU