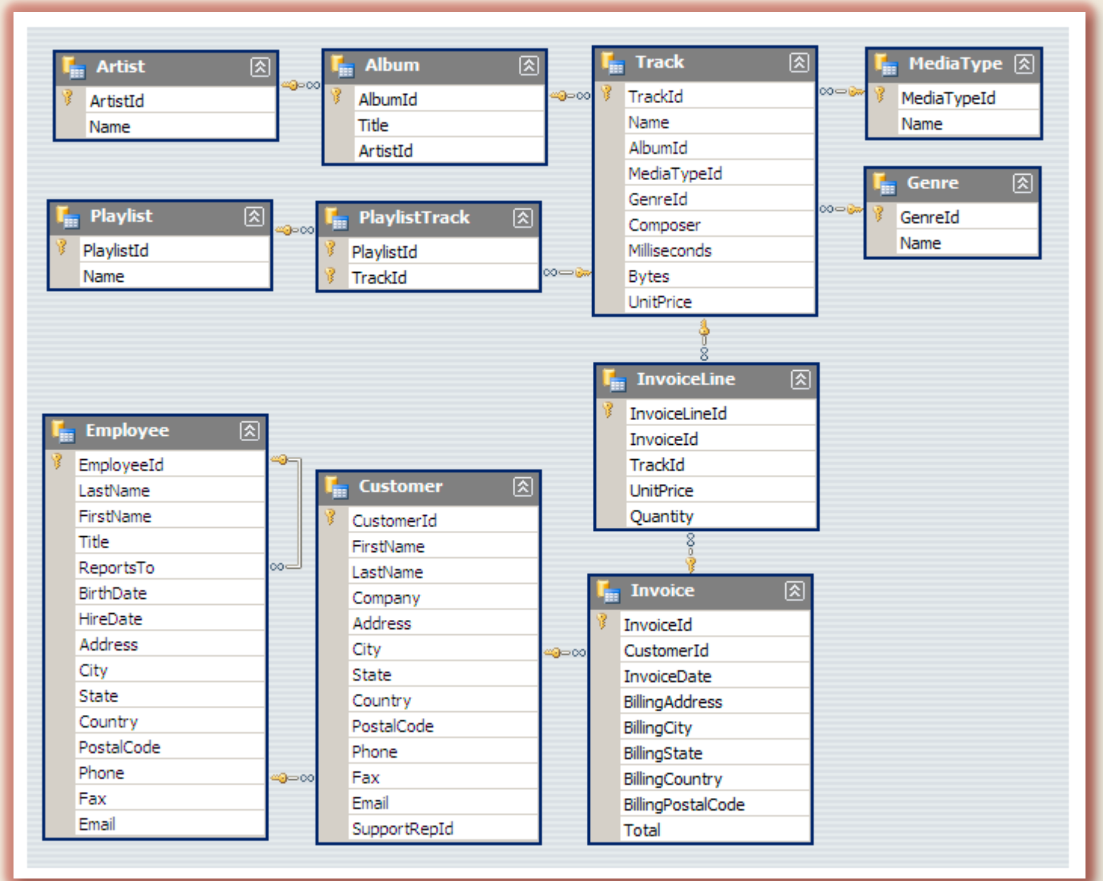
Music Store Analysis Using SQL

Giri Sai Manikanta Dara

Questions Raised :

- Q1: Who is the senior most employee based on job title?
- Q2: Which countries have the most Invoices? only top 5 countries
- Q3: What are top 3 values of total invoice?
- 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
- Q5: Who are the top 5 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.
- 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.
- 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 5 rock bands.
- 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.
- Q9: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent?
- 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.
- 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.

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

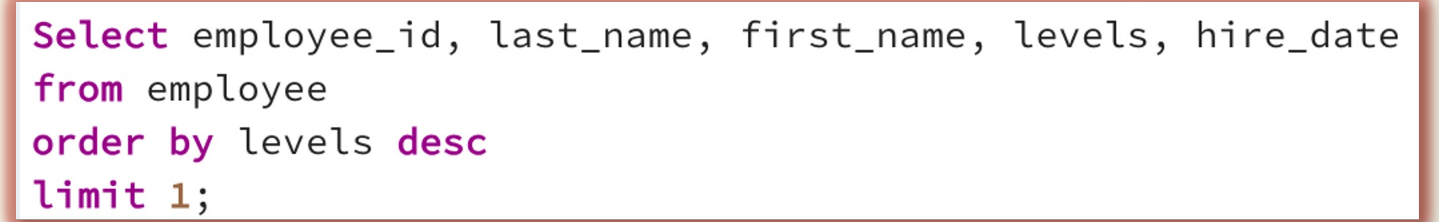
Employee
- EmployeeId
- LastName
- FirstName
- Title
- ReportsTo
- BirthDate
- HireDate
- Address
- City
- State
- Country
- PostalCode
- Phone
- Fax
- Email

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

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

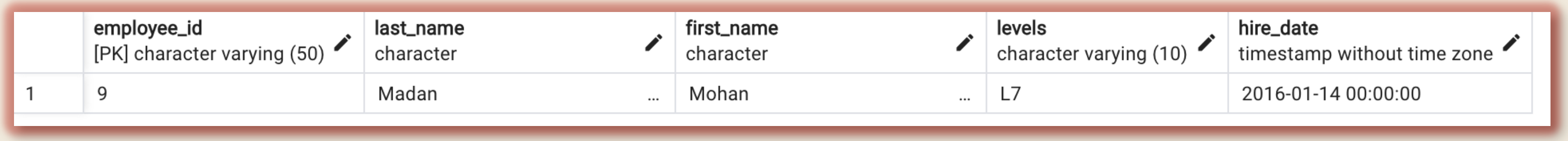
♫ Q1: Who is the senior most employee based on job title

Query:

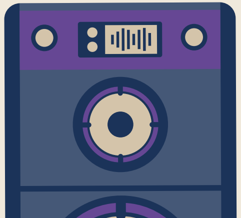


```
Select employee_id, last_name, first_name, levels, hire_date
from employee
order by levels desc
limit 1;
```

Data Output:

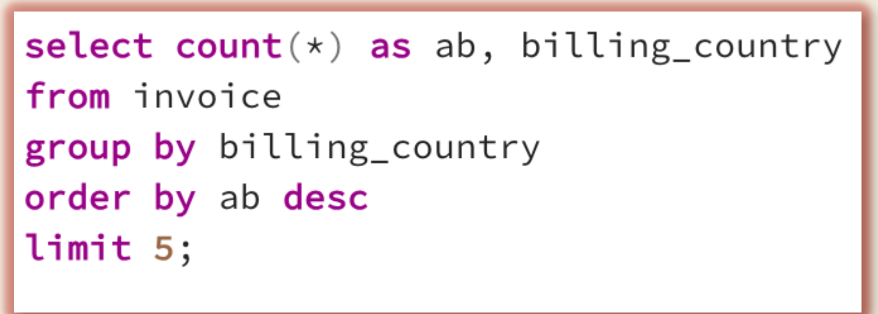


	employee_id [PK] character varying (50)	last_name character	first_name character	levels character varying (10)	hire_date timestamp without time zone
1	9	Madan ...	Mohan ...	L7	2016-01-14 00:00:00



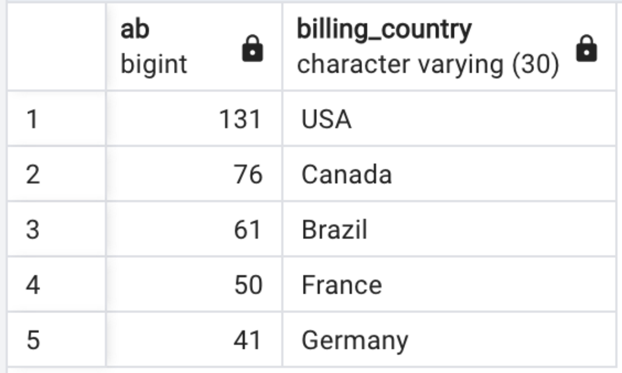
Q2: Which countries have the most Invoices? only top 5 countries

Query:



```
select count(*) as ab, billing_country
from invoice
group by billing_country
order by ab desc
limit 5;
```

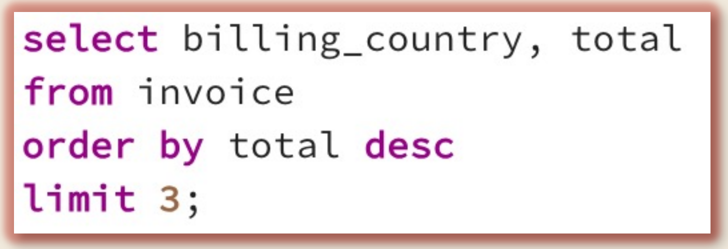
Data Output:



	ab bigint 🔒	billing_country character varying (30) 🔒
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany

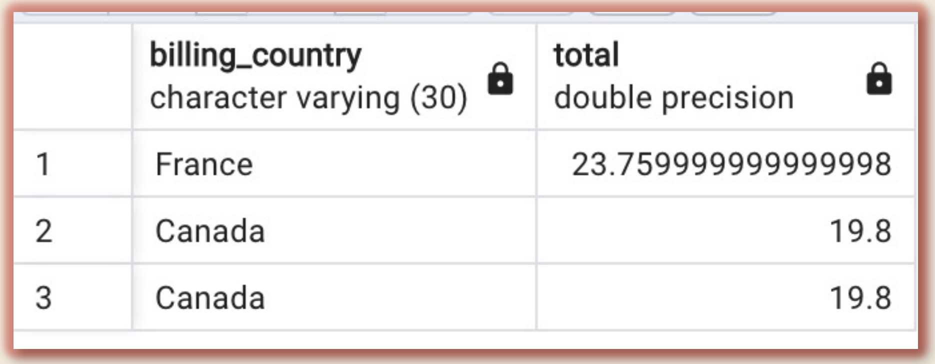
Q3: What are top 3 values of total invoice?

Query:

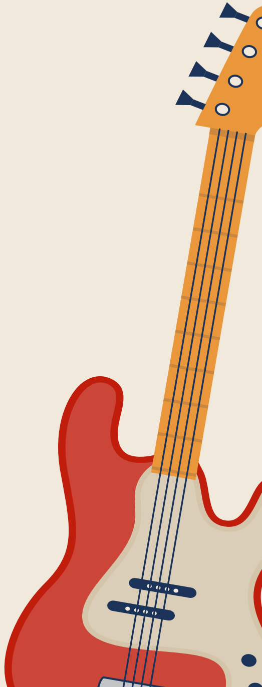


```
select billing_country, total
from invoice
order by total desc
limit 3;
```

Data Output:

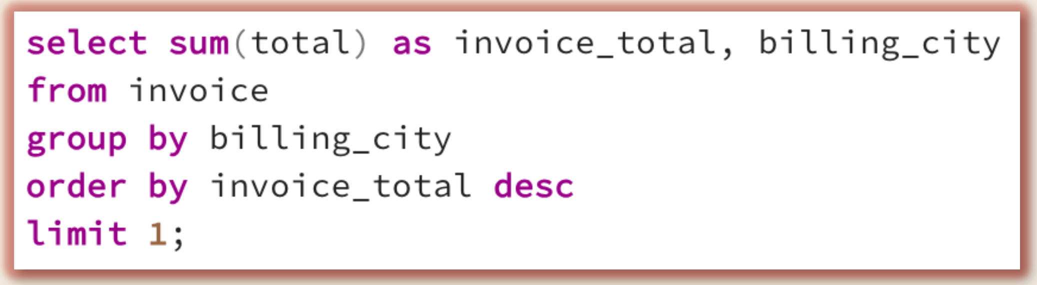


	billing_country character varying (30) 🔒	total double precision 🔒
1	France	23.759999999999998
2	Canada	19.8
3	Canada	19.8



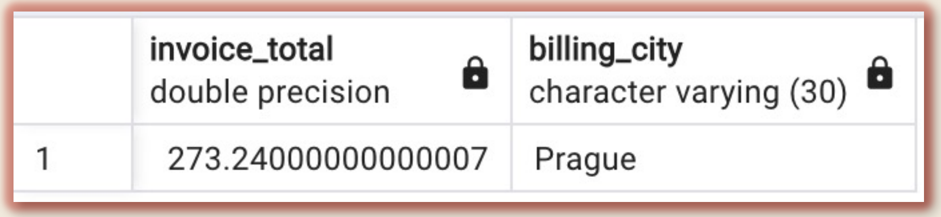
Q4: Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals

Query:



```
select sum(total) as invoice_total, billing_city
from invoice
group by billing_city
order by invoice_total desc
limit 1;
```

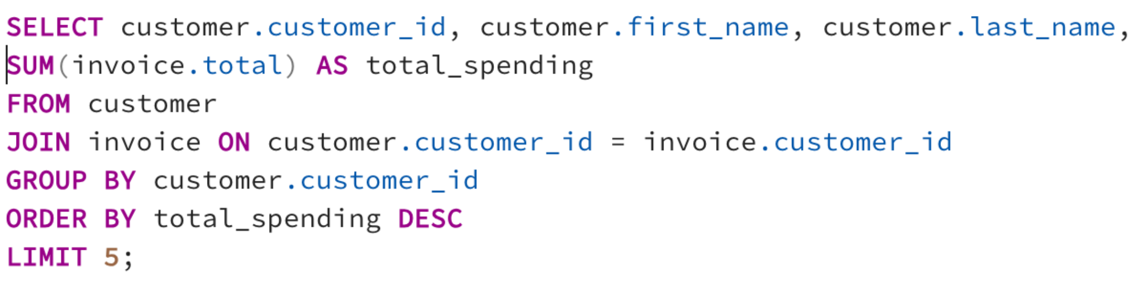
Data Output:



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

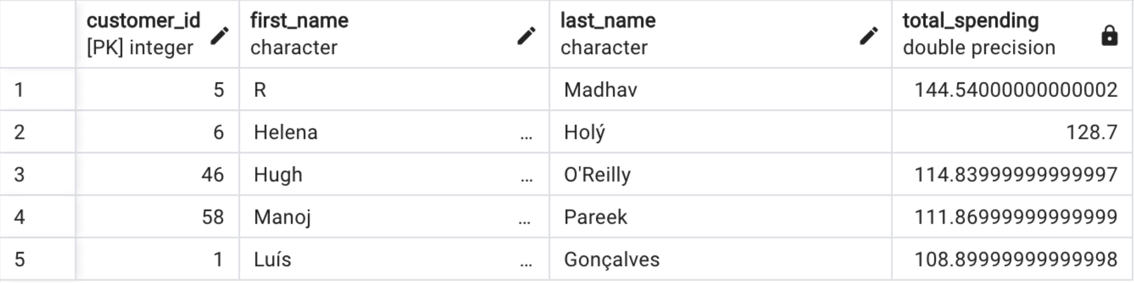
Q5: Who are the top 5 best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money.

Query:



```
SELECT customer.customer_id, customer.first_name, customer.last_name,
SUM(invoice.total) AS total_spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total_spending DESC
LIMIT 5;
```

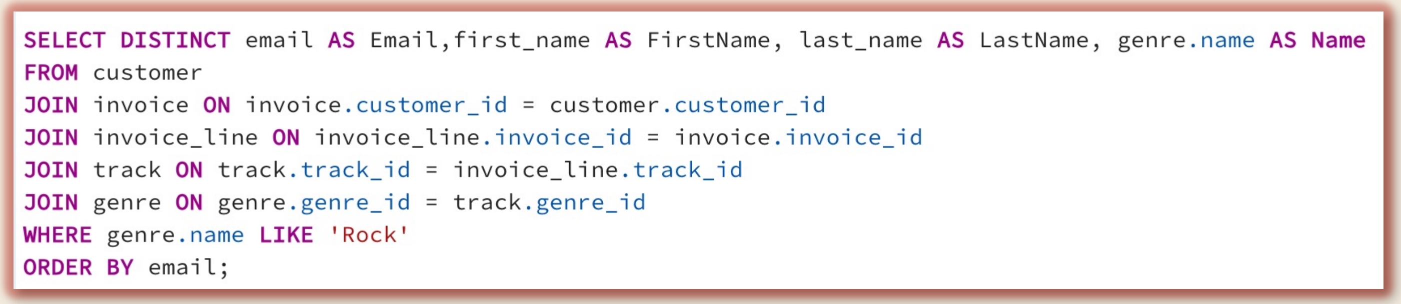
Data Output:



	customer_id [PK] integer	first_name character	last_name character	total_spending double precision
1	5	R	Madhav	144.54000000000002
2	6	Helena ...	Holý	128.7
3	46	Hugh ...	O'Reilly	114.83999999999997
4	58	Manoj ...	Pareek	111.86999999999999
5	1	Luís ...	Gonçalves	108.89999999999998

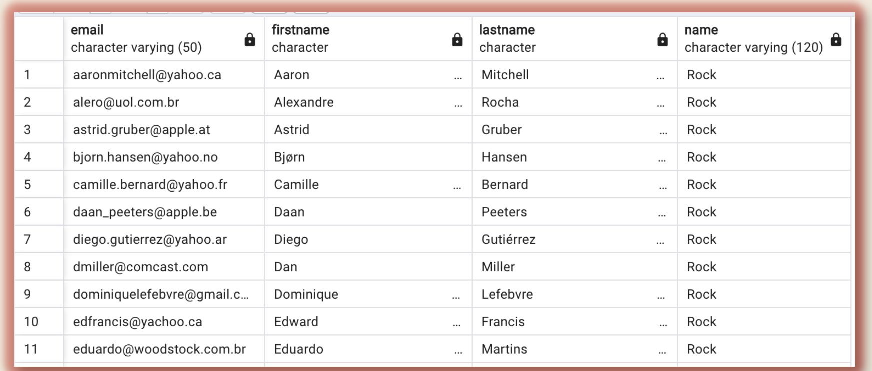
Q6: Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A.

Query:

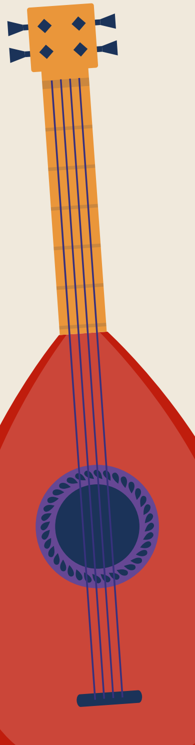


```sql
SELECT DISTINCT email AS Email,first_name AS FirstName, last_name AS LastName, genre.name AS Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

Data Output:



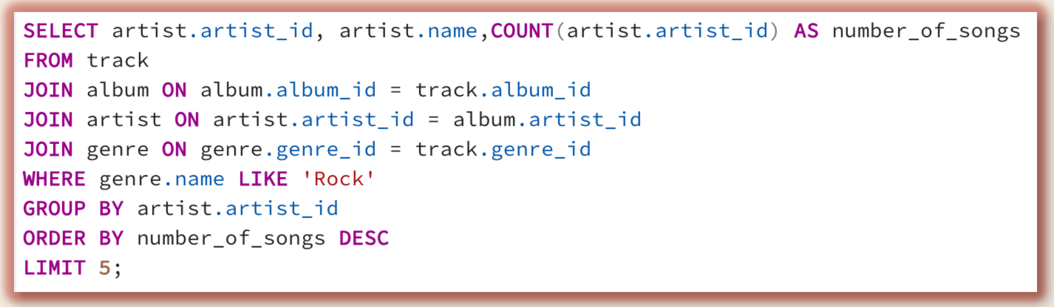
	email character varying (50)	firstname character	lastname character	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid.gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock
6	daan_peeters@apple.be	Daan	Peeters	Rock
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock
8	dmiller@comcast.com	Dan	Miller	Rock
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre	Rock
10	edfrancis@yachoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock



@manikanta1801

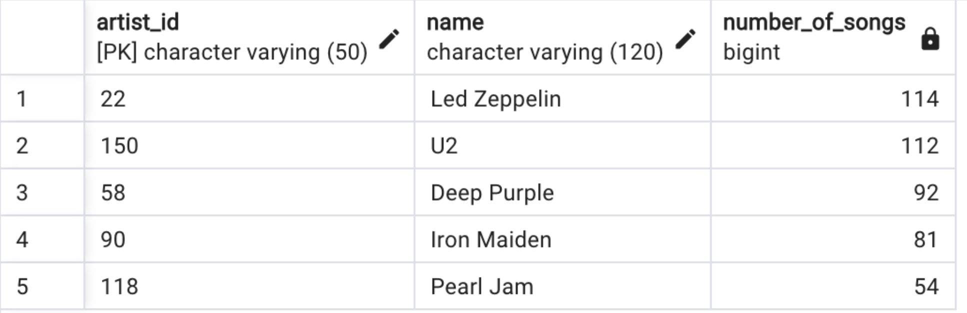
Q7: Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 5 rock bands.

Query:



```sql
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 LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs DESC
LIMIT 5;
```

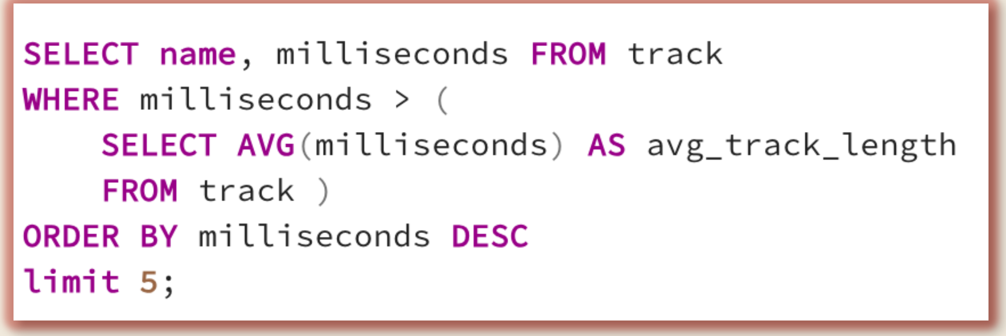
Data Output:



	artist_id [PK] character varying (50)	name character varying (120)	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

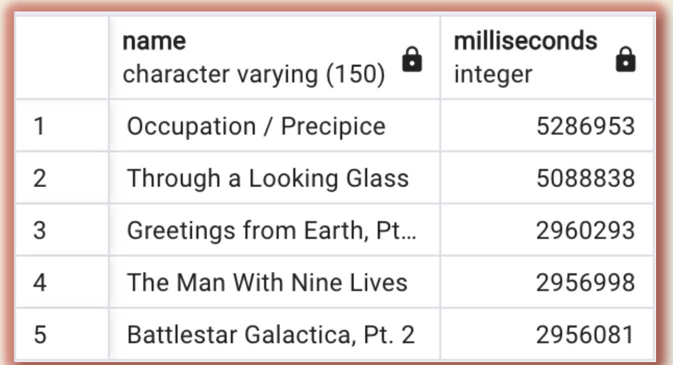
Q8: Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

Query:

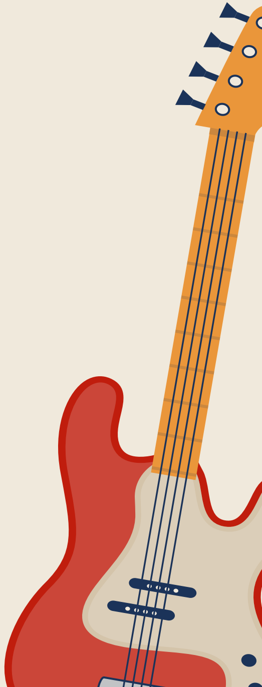


```
SELECT name, milliseconds FROM track
WHERE milliseconds > (
    SELECT AVG(milliseconds) AS avg_track_length
    FROM track )
ORDER BY milliseconds DESC
limit 5;
```

Data Output:

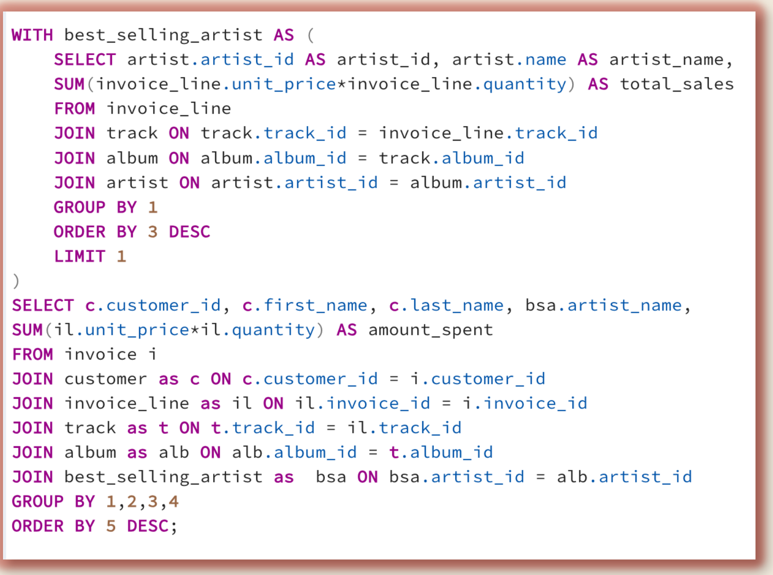


	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt...	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081



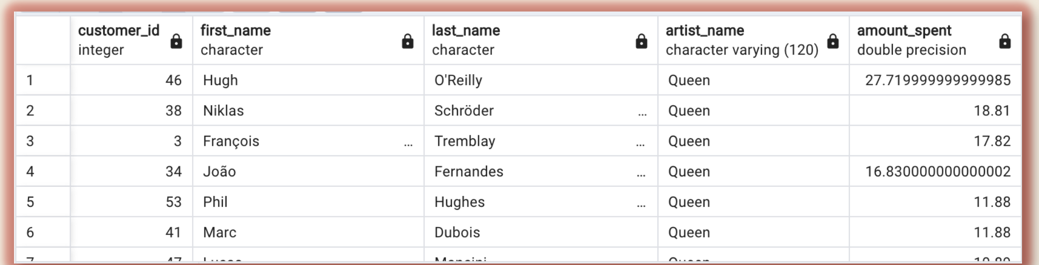
Q9: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

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 as c ON c.customer_id = i.customer_id
JOIN invoice_line as il ON il.invoice_id = i.invoice_id
JOIN track as t ON t.track_id = il.track_id
JOIN album as alb ON alb.album_id = t.album_id
JOIN best_selling_artist as  bsa ON bsa.artist_id = alb.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC;
```

Data Output:

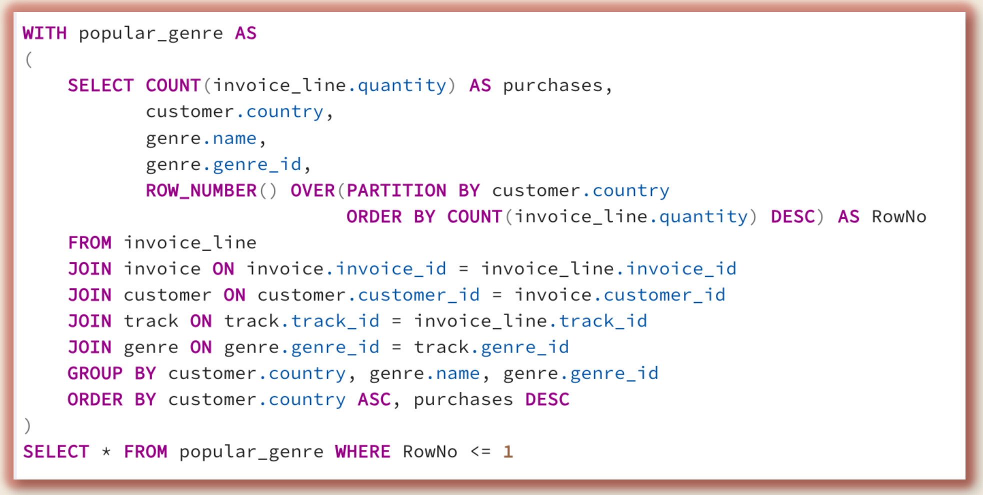


	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder ...	Queen	18.81
3	3	François ...	Tremblay ...	Queen	17.82
4	34	João	Fernandes ...	Queen	16.830000000000002
5	53	Phil	Hughes ...	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89

To solve the problem, first, identify the top-earning artist by joining the `Invoice_Line`, `Track`, `Album`, and `Artist` tables, then calculate total earnings for each artist by summing `Unit_Price * Quantity` and selecting the artist with the highest earnings. Next, find the customer who spent the most on this artist by joining the `Invoice_Line`, `Invoice`, `Track`, `Album`, `Artist`, and `Customer` tables, filtering for the identified artist, calculating total spending per customer by summing `Unit_Price * Quantity`, and selecting the customer with the highest spending on this artist.

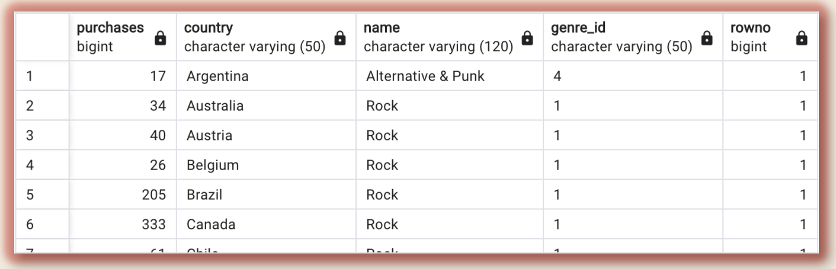
Q10: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres.

Query:



```sql
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 customer.country, genre.name, genre.genre_id
    ORDER BY customer.country ASC, purchases DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1
```

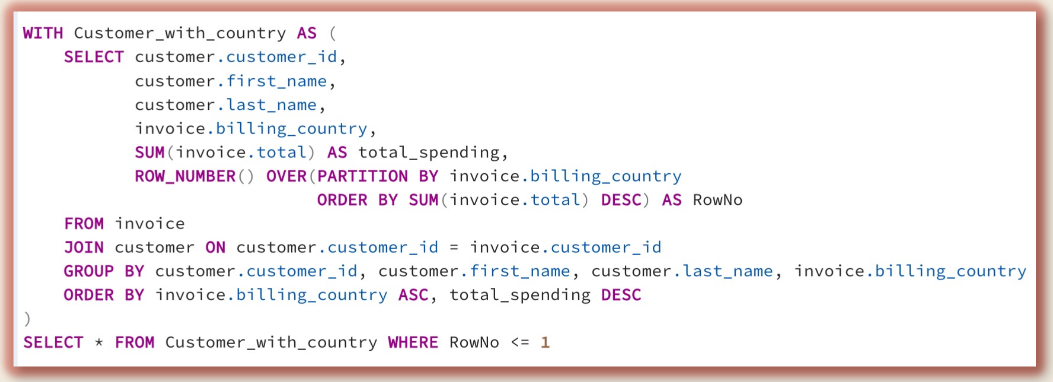
Data Output:



	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

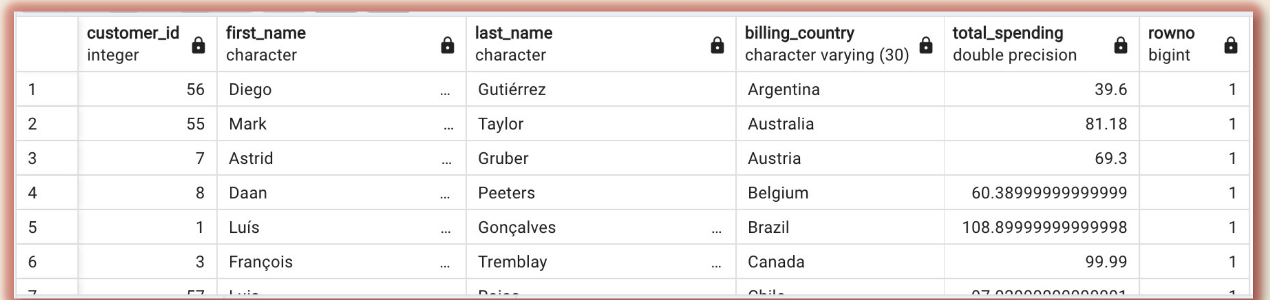
Q11: Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.

Query:



```
WITH Customer_with_country AS (
    SELECT customer.customer_id,
           customer.first_name,
           customer.last_name,
           invoice.billing_country,
           SUM(invoice.total) AS total_spending,
           ROW_NUMBER() OVER(PARTITION BY invoice.billing_country
                             ORDER BY SUM(invoice.total) DESC) AS RowNo
    FROM invoice
    JOIN customer ON customer.customer_id = invoice.customer_id
    GROUP BY customer.customer_id, customer.first_name, customer.last_name, invoice.billing_country
    ORDER BY invoice.billing_country ASC, total_spending DESC
)
SELECT * FROM Customer_with_country WHERE RowNo <= 1
```

Data 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

Thank You

Giri Sai Manikanta Dara

@manikanta1801

daragirisaimanikanta.netlify.app