MUSIC STORE

MySQL ANALYSIS

Presented By

Harshit Bisht

INTRODUCTION

This project focuses on analyzing sales data for a music store using MySQL.

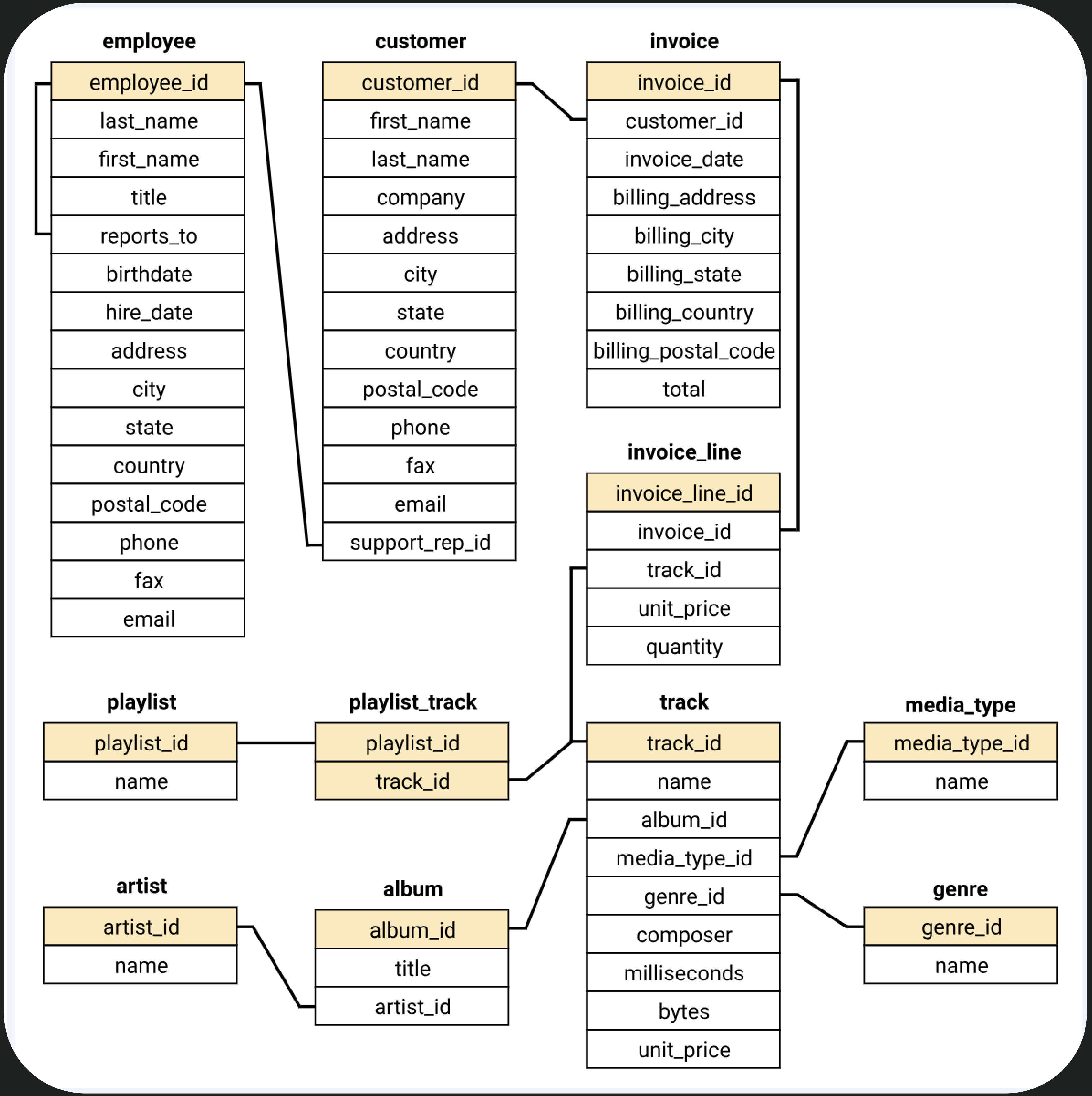
By examining key metrics such as revenue, product categories, customer demographics, and sales trends, the analysis provides insights into the store's performance.

The goal is to identify patterns, optimize inventory, and improve sales strategies.

MySQL queries were employed to retrieve, filter, and aggregate data from various tables, enabling detailed reporting on sales by artist, genre, and region.

This analysis aids in making data-driven decisions to enhance business operations and customer satisfaction.

ENTITY RELATIONSHIP DIAGRAM



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

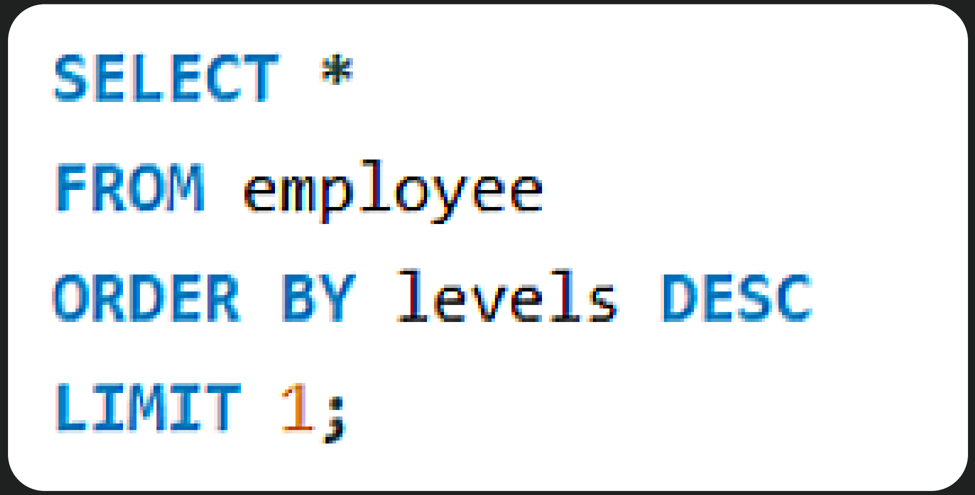
media_type
media_type_id
name

artist
artist_id
name

album
album_id
title
artist_id

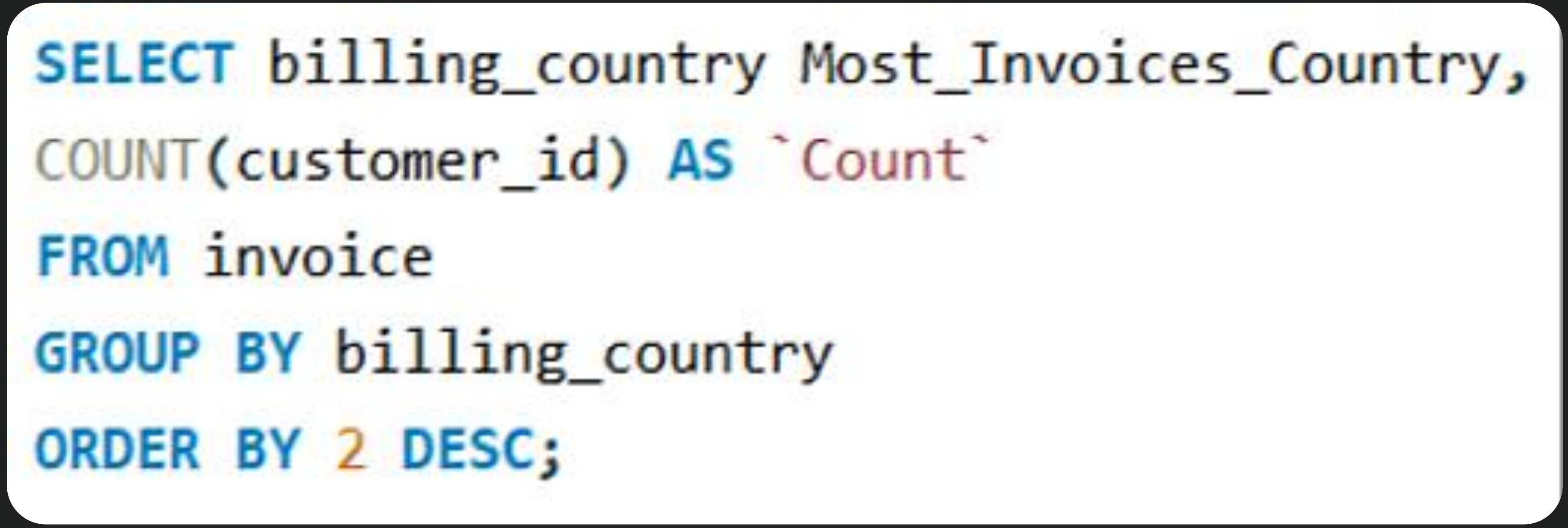
genre
genre_id
name

1 Who is the senior most employee based on job title?



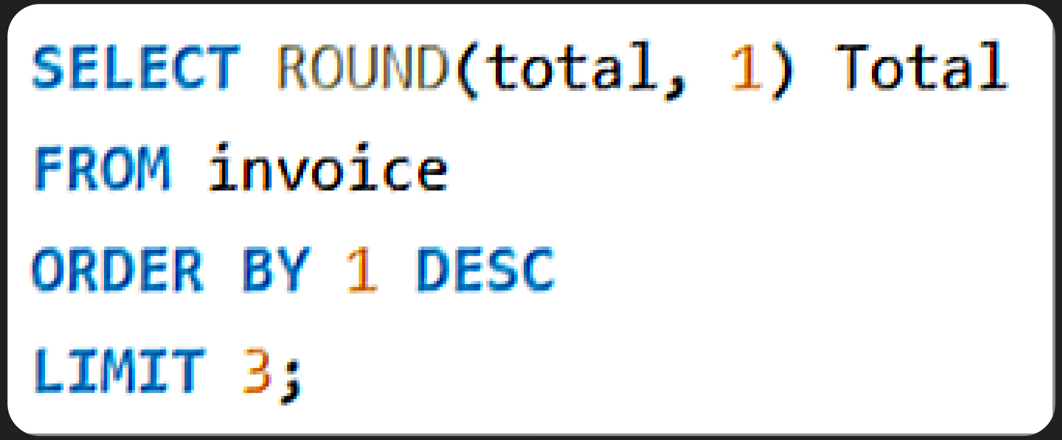
```
SELECT *
FROM employee
ORDER BY levels DESC
LIMIT 1;
```

2 Which countries have the most Invoices?



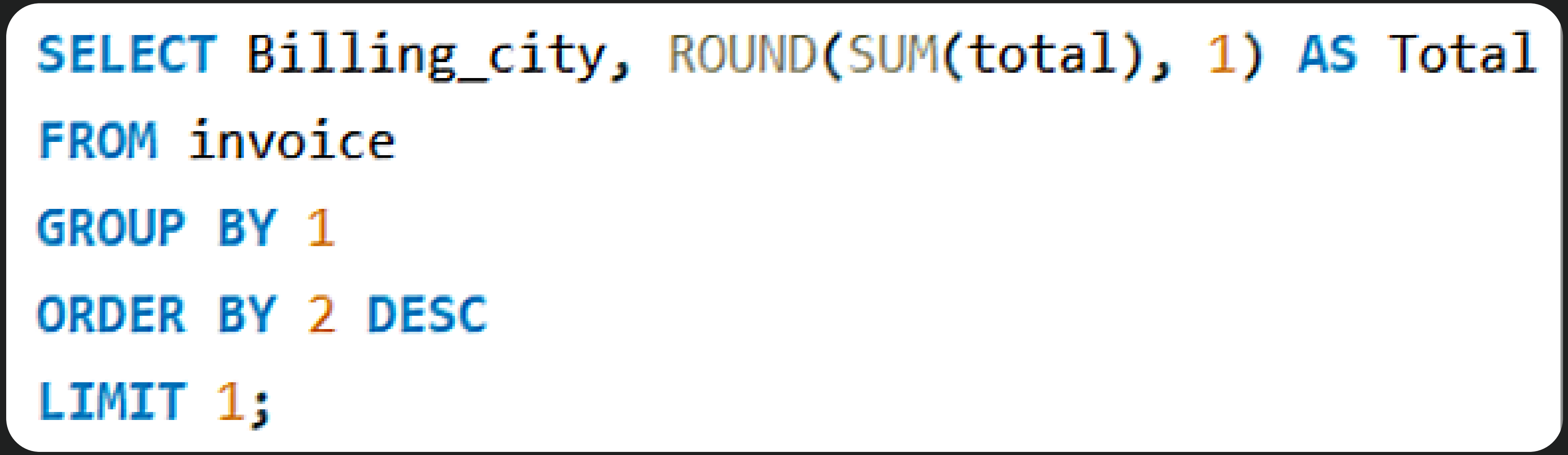
```
SELECT billing_country Most_Invoices_Country,
COUNT(customer_id) AS `Count`
FROM invoice
GROUP BY billing_country
ORDER BY 2 DESC;
```

3 What are top 3 values of total invoice?



```
SELECT ROUND(total, 1) Total
FROM invoice
ORDER BY 1 DESC
LIMIT 3;
```

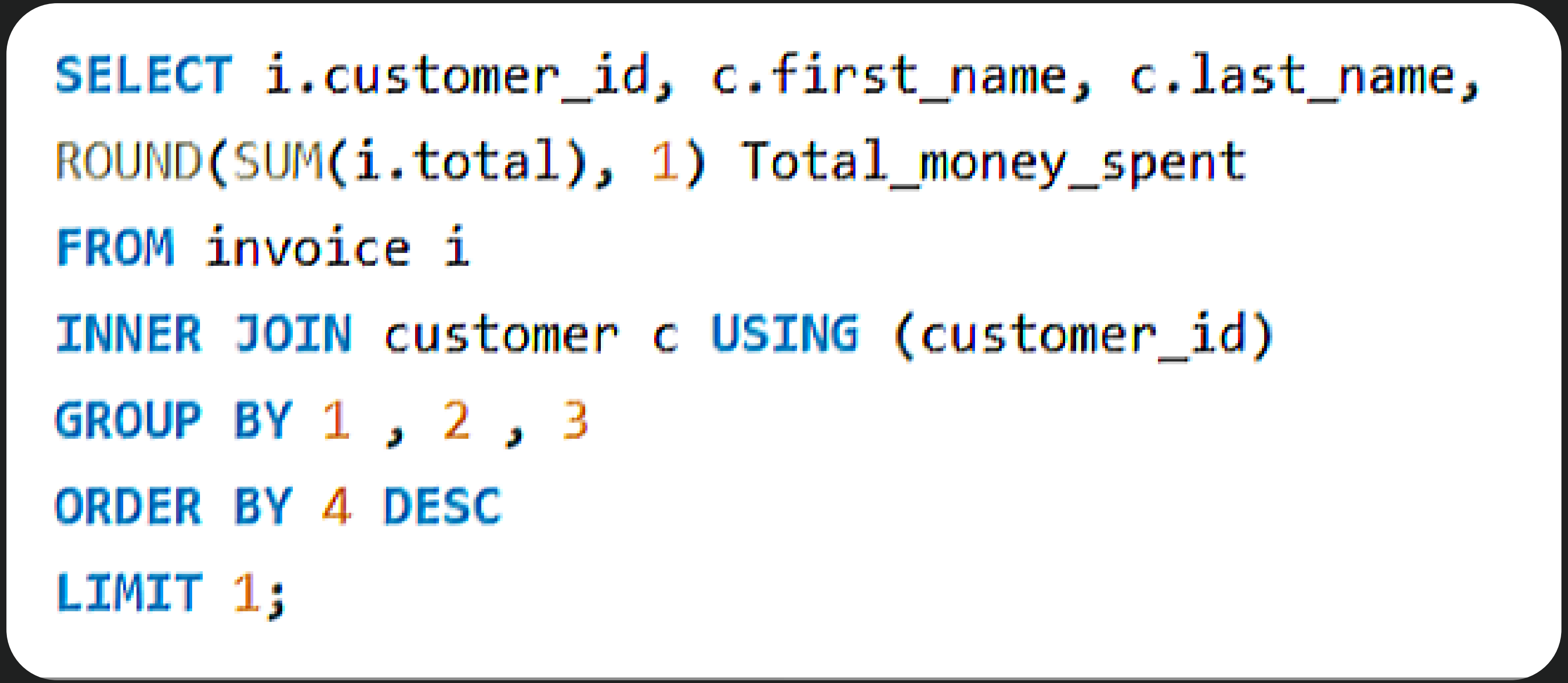
4 Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money.



```
SELECT Billing_city, ROUND(SUM(total), 1) AS Total
FROM invoice
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

5 Who is the best customer?

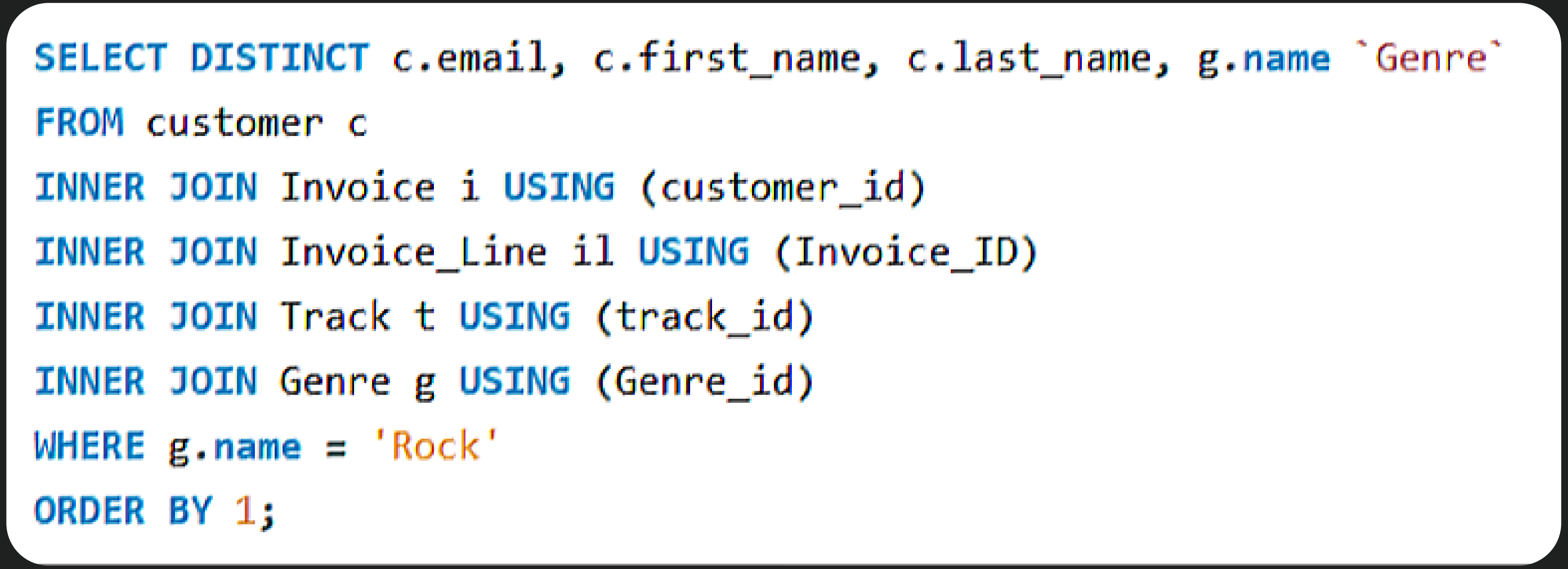
The customer who has spent the most money will be declared the best customer.



```
SELECT i.customer_id, c.first_name, c.last_name,
ROUND(SUM(i.total), 1) Total_money_spent
FROM invoice i
INNER JOIN customer c USING (customer_id)
GROUP BY 1 , 2 , 3
ORDER BY 4 DESC
LIMIT 1;
```

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

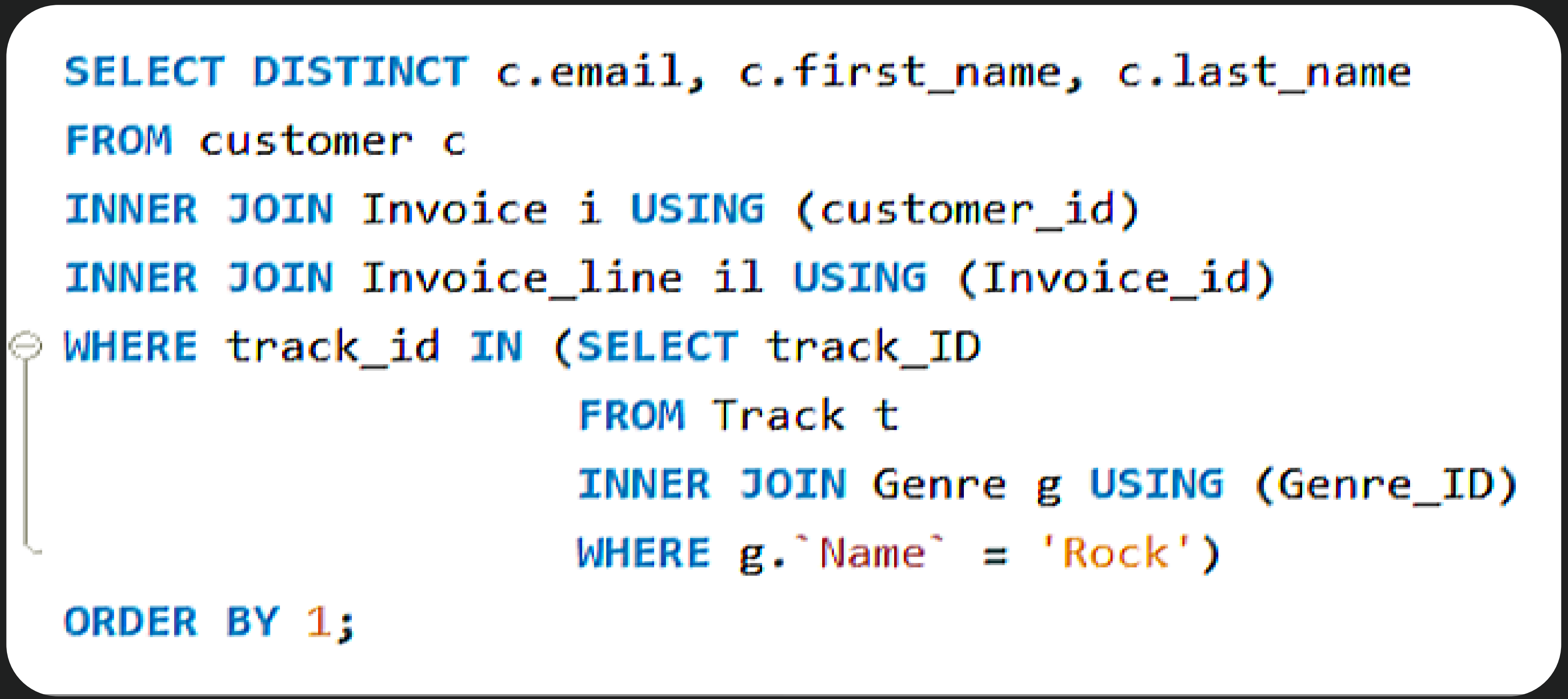
Return your list ordered alphabetically by email starting with A



```sql
SELECT DISTINCT c.email, c.first_name, c.last_name, g.name `Genre`
FROM customer c
INNER JOIN Invoice i USING (customer_id)
INNER JOIN Invoice_Line il USING (Invoice_ID)
INNER JOIN Track t USING (track_id)
INNER JOIN Genre g USING (Genre_id)
WHERE g.name = 'Rock'
ORDER BY 1;
```

Approach

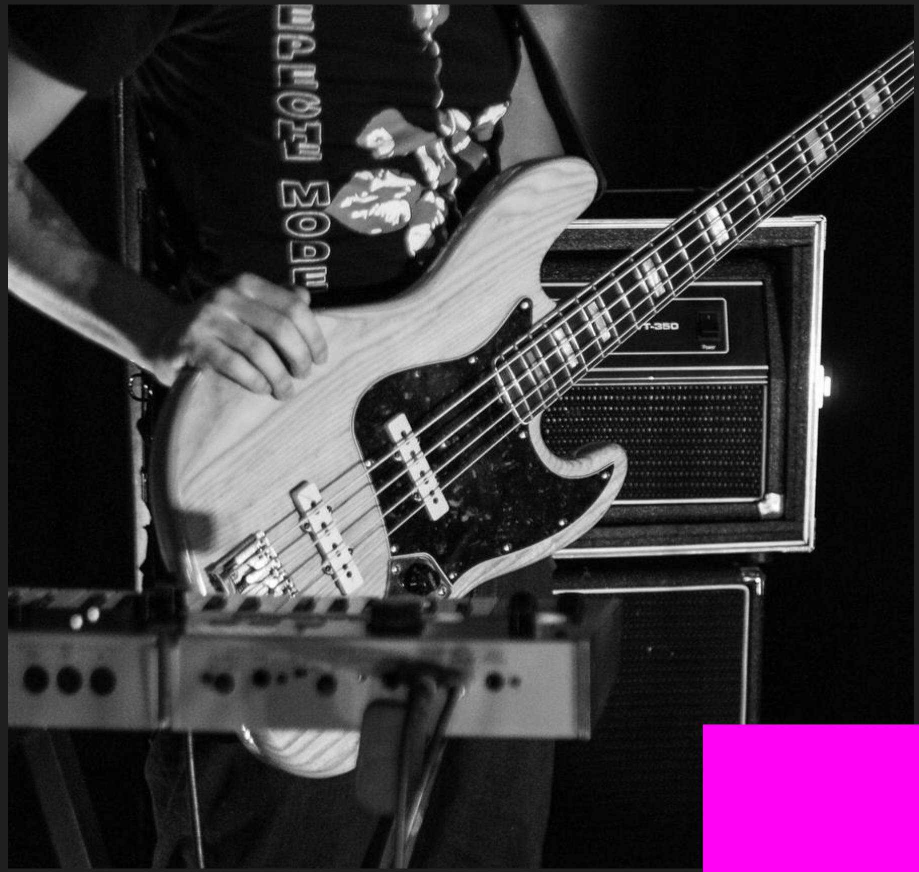
1



```sql
SELECT DISTINCT c.email, c.first_name, c.last_name
FROM customer c
INNER JOIN Invoice i USING (customer_id)
INNER JOIN Invoice_line il USING (Invoice_id)
WHERE track_id IN (SELECT track_ID
                   FROM Track t
                   INNER JOIN Genre g USING (Genre_ID)
                   WHERE g.`Name` = 'Rock')
ORDER BY 1;
```

Approach

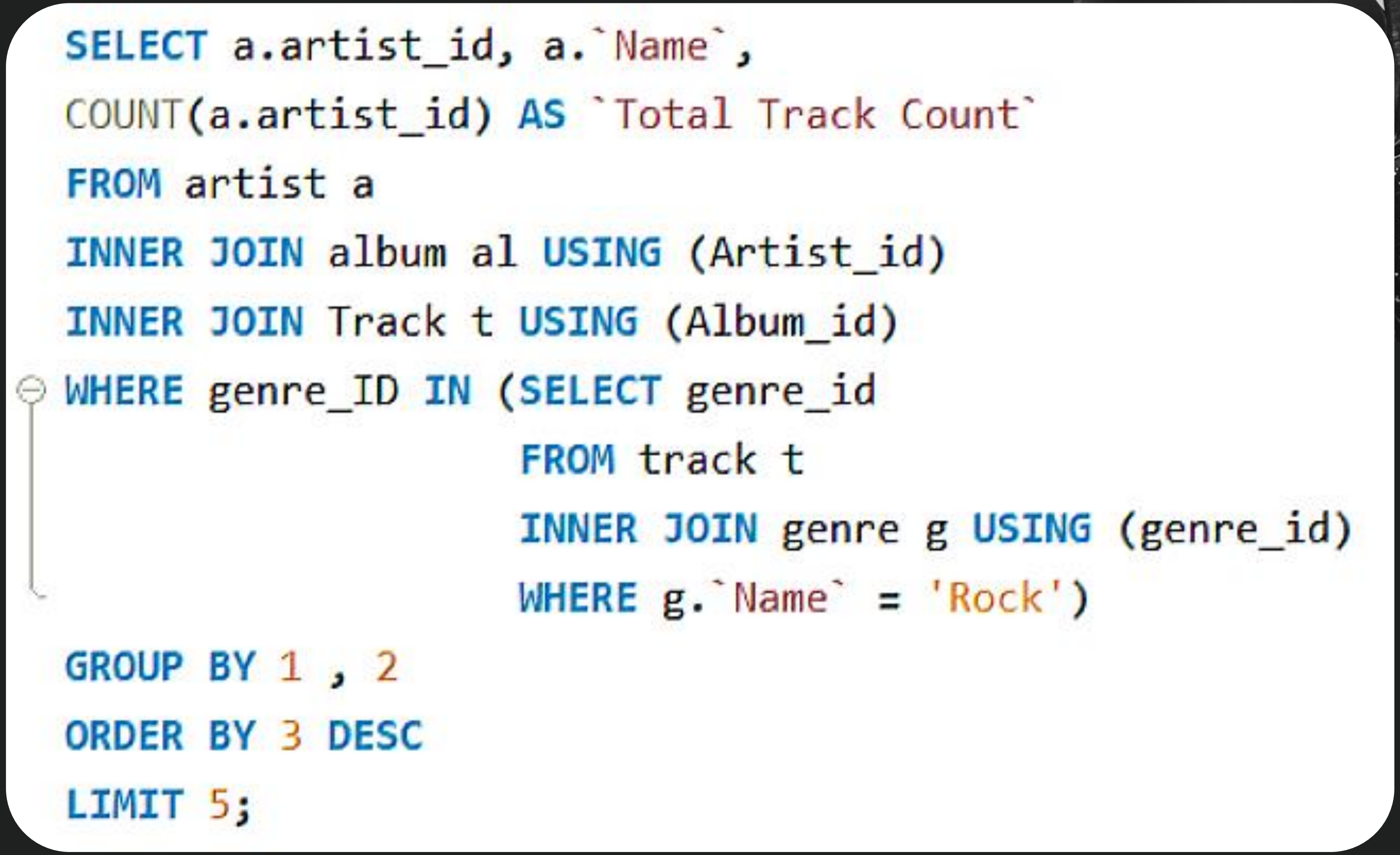
2



7 Let's invite the artists who have written the most rock music in our dataset.

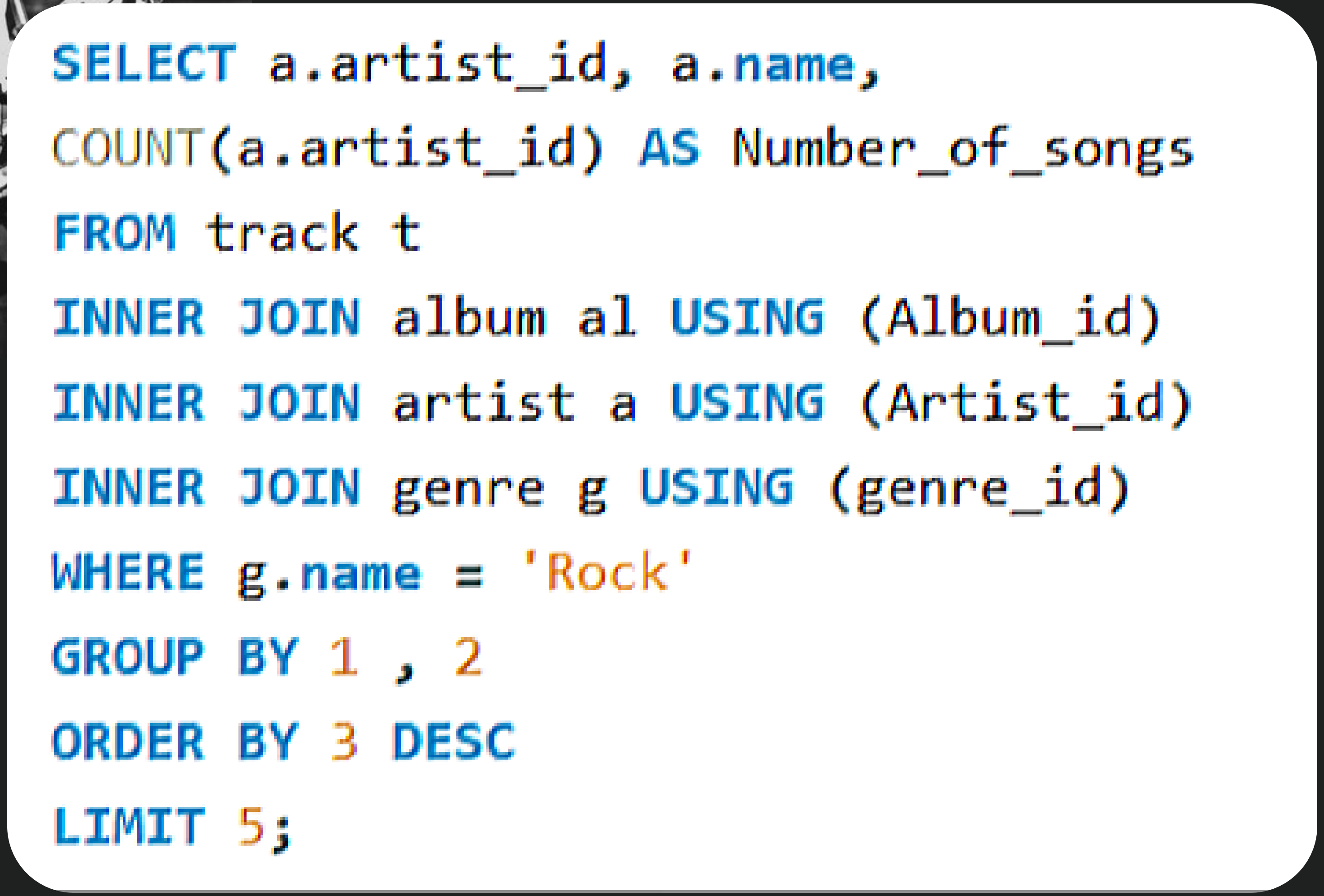
Write a query that returns the Artist name and total track count of the top 5 rock bands

Approach 1



```sql
SELECT a.artist_id, a.`Name`,
COUNT(a.artist_id) AS `Total Track Count`
FROM artist a
INNER JOIN album al USING (Artist_id)
INNER JOIN Track t USING (Album_id)
WHERE genre_ID IN (SELECT genre_id
                   FROM track t
                   INNER JOIN genre g USING (genre_id)
                   WHERE g.`Name` = 'Rock')
GROUP BY 1 , 2
ORDER BY 3 DESC
LIMIT 5;
```

Approach 2

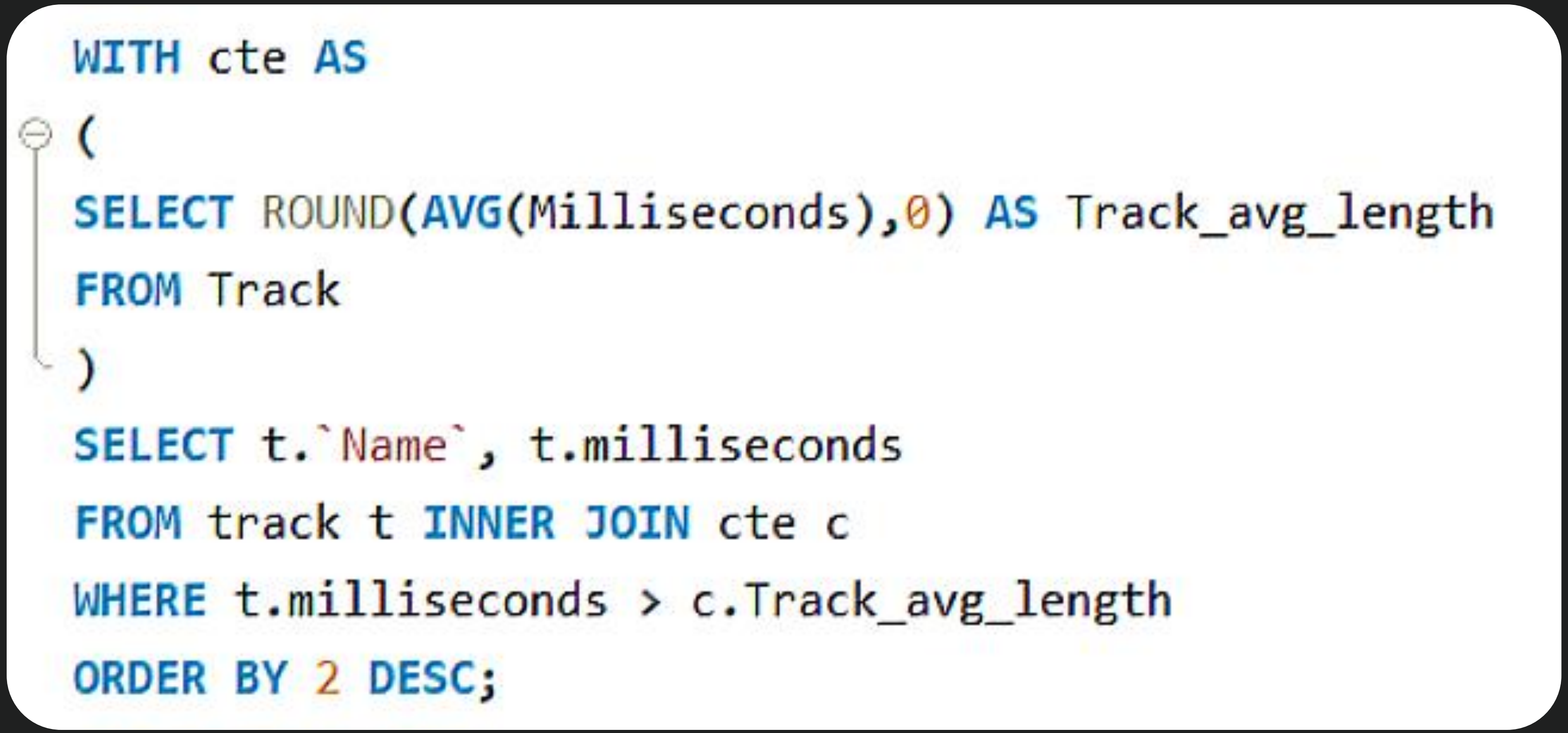


```sql
SELECT a.artist_id, a.name,
COUNT(a.artist_id) AS Number_of_songs
FROM track t
INNER JOIN album al USING (Album_id)
INNER JOIN artist a USING (Artist_id)
INNER JOIN genre g USING (genre_id)
WHERE g.name = 'Rock'
GROUP BY 1 , 2
ORDER BY 3 DESC
LIMIT 5;
```

8 **Return all the track names that have a song length longer than the average song length.**

Return the Name and Milliseconds for each track.

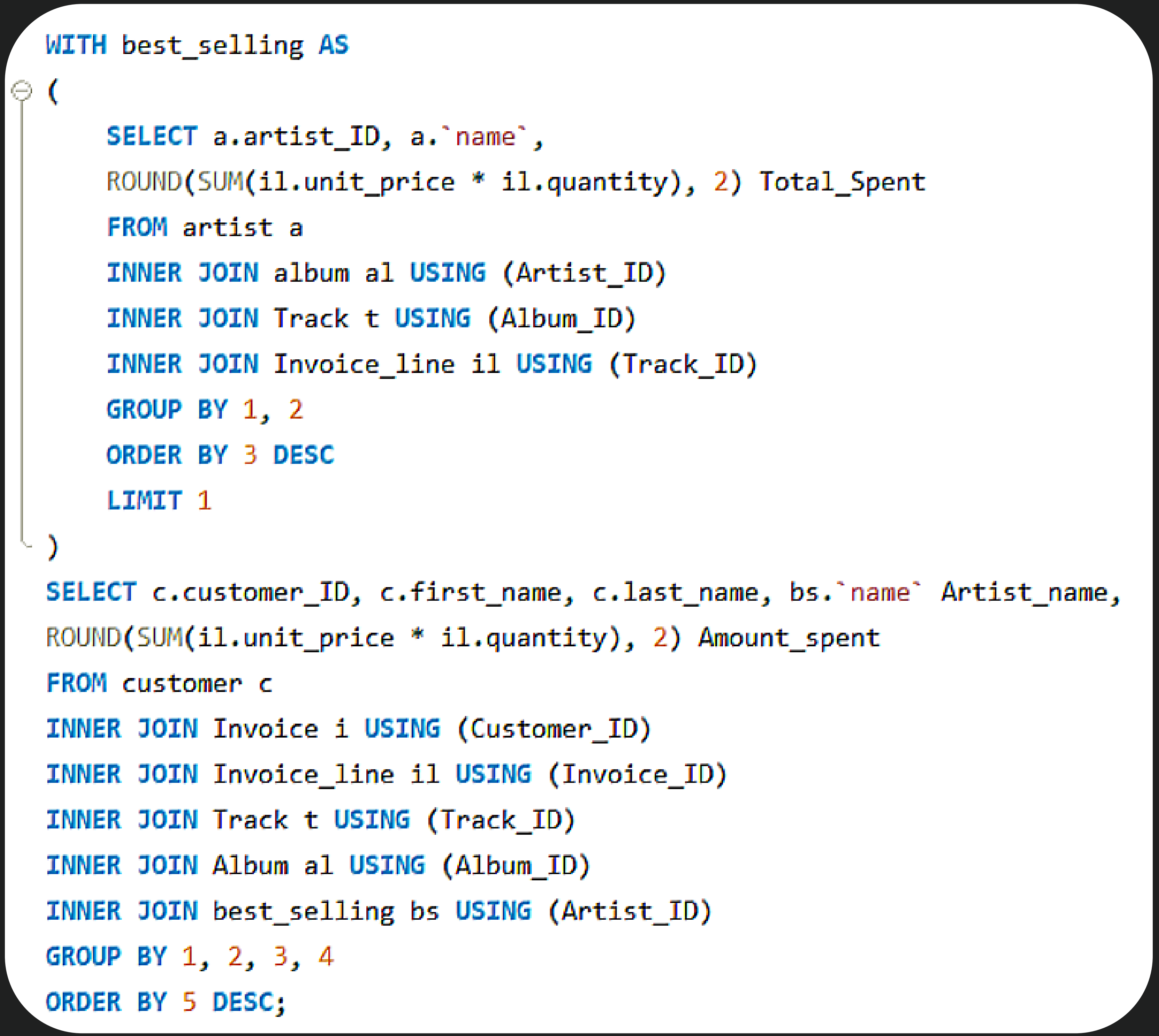
Order by the song length with the longest songs listed first



```
WITH cte AS
(
SELECT ROUND(AVG(Milliseconds),0) AS Track_avg_length
FROM Track
)
SELECT t.`Name`, t.milliseconds
FROM track t INNER JOIN cte c
WHERE t.milliseconds > c.Track_avg_length
ORDER BY 2 DESC;
```

9 Find how much amount spent by each customer on the best selling artists?

Write a query to return customer ID, customer name, artist name and total spent



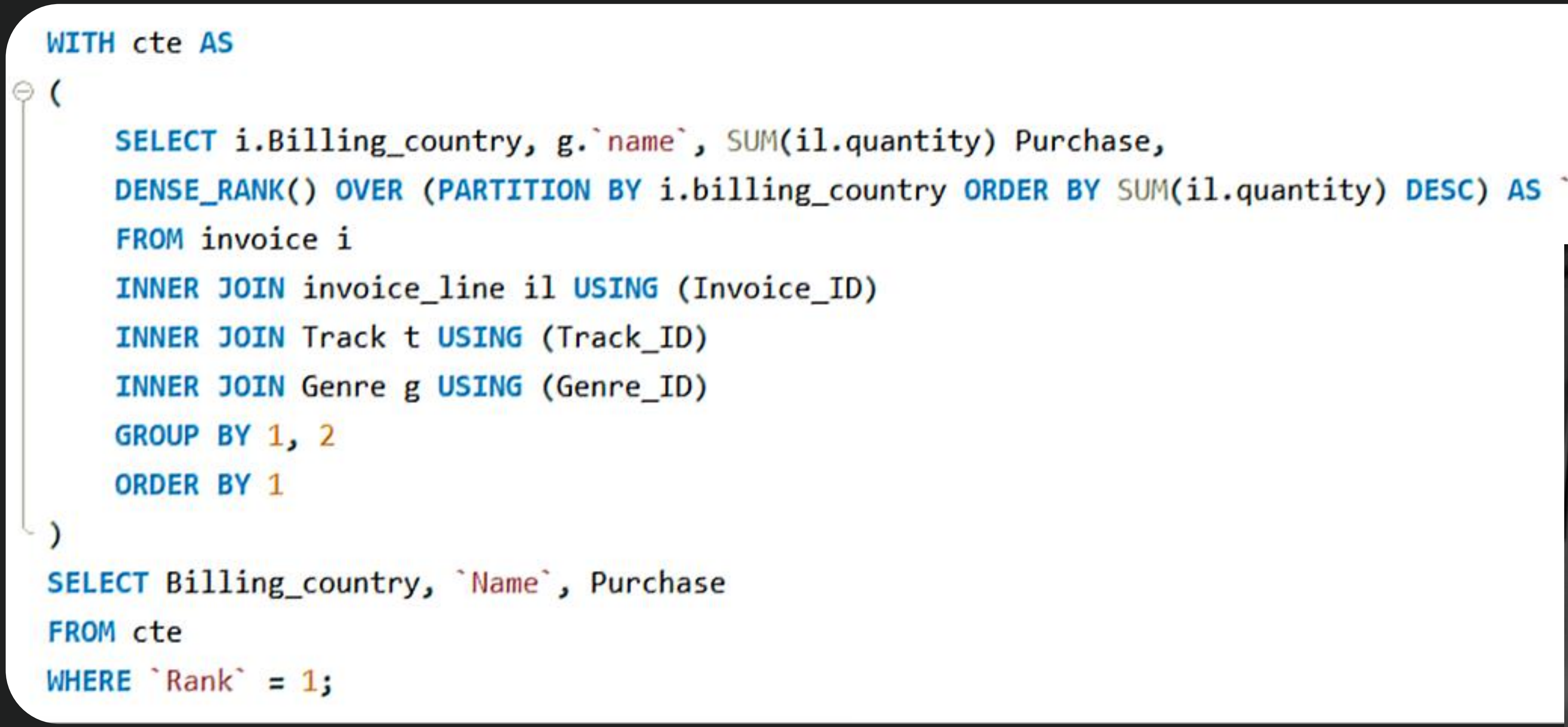
```
WITH best_selling AS
(
    SELECT a.artist_ID, a.`name`,
    ROUND(SUM(il.unit_price * il.quantity), 2) Total_Spent
    FROM artist a
    INNER JOIN album al USING (Artist_ID)
    INNER JOIN Track t USING (Album_ID)
    INNER JOIN Invoice_line il USING (Track_ID)
    GROUP BY 1, 2
    ORDER BY 3 DESC
    LIMIT 1
)
SELECT c.customer_ID, c.first_name, c.last_name, bs.`name` Artist_name,
ROUND(SUM(il.unit_price * il.quantity), 2) Amount_spent
FROM customer c
INNER JOIN Invoice i USING (Customer_ID)
INNER JOIN Invoice_line il USING (Invoice_ID)
INNER JOIN Track t USING (Track_ID)
INNER JOIN Album al USING (Album_ID)
INNER JOIN best_selling bs USING (Artist_ID)
GROUP BY 1, 2, 3, 4
ORDER BY 5 DESC;
```

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



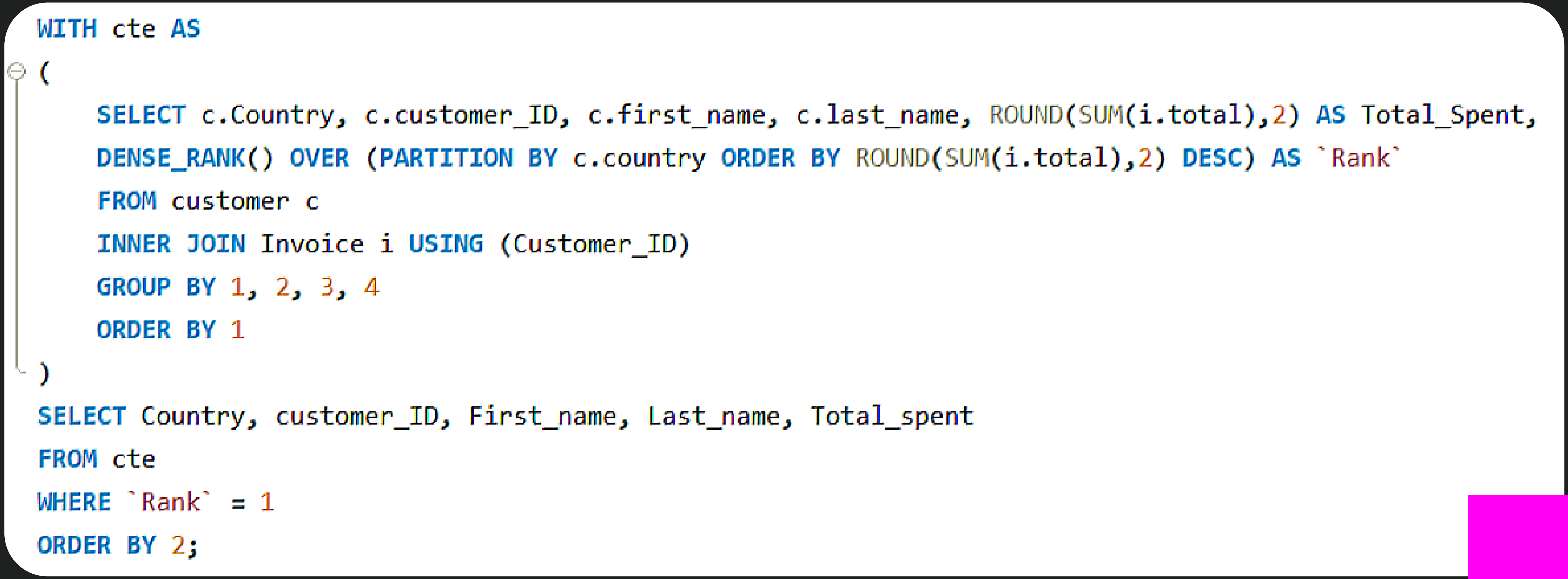
```
WITH cte AS
(
    SELECT i.Billing_country, g.`name`, SUM(il.quantity) Purchase,
    DENSE_RANK() OVER (PARTITION BY i.billing_country ORDER BY SUM(il.quantity) DESC) AS `Rank`
    FROM invoice i
    INNER JOIN invoice_line il USING (Invoice_ID)
    INNER JOIN Track t USING (Track_ID)
    INNER JOIN Genre g USING (Genre_ID)
    GROUP BY 1, 2
    ORDER BY 1
)
SELECT Billing_country, `Name`, Purchase
FROM cte
WHERE `Rank` = 1;
```

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 cte AS
(
    SELECT c.Country, c.customer_ID, c.first_name, c.last_name, ROUND(SUM(i.total),2) AS Total_Spent,
    DENSE_RANK() OVER (PARTITION BY c.country ORDER BY ROUND(SUM(i.total),2) DESC) AS `Rank`
    FROM customer c
    INNER JOIN Invoice i USING (Customer_ID)
    GROUP BY 1, 2, 3, 4
    ORDER BY 1
)
SELECT Country, customer_ID, First_name, Last_name, Total_spent
FROM cte
WHERE `Rank` = 1
ORDER BY 2;
```

SUMMARY

Multichannel Integration: Create a seamless shopping experience with options like in-store pickups and easy returns for online purchases.

Leverage Data: Utilize sales data to tailor marketing efforts and promotions based on customer preferences and buying trends.

Optimize Online Presence: Enhance website usability with a mobile-friendly design, secure checkout, and strong SEO practices to drive organic traffic.

Social Media Engagement: Actively engage customers on social media by showcasing new products, running contests, and sharing user-generated content.

Email Marketing Campaigns: Build a robust email list to send personalized offers, updates on new arrivals, and event invitations to keep customers connected.

In-Store Experiences: Offer live events, workshops, and personalized recommendations to create an engaging atmosphere that encourages visits.

Bundling Products: Create attractive bundles or package deals (e.g., instrument + accessories) to increase average order value and enhance customer satisfaction.

Customer Feedback Loop: Regularly seek customer feedback to understand their needs better and adapt offerings accordingly.

Seasonal Promotions: Implement timely promotions tied to holidays or special events to attract customers and boost sales.

Flexible Payment Options: Provide various payment methods, including financing for high-ticket items, to cater to diverse customer preferences.

THANK YOU
FOR READING

linkedin.com/in/harshit-bisht-249735216

MySQL