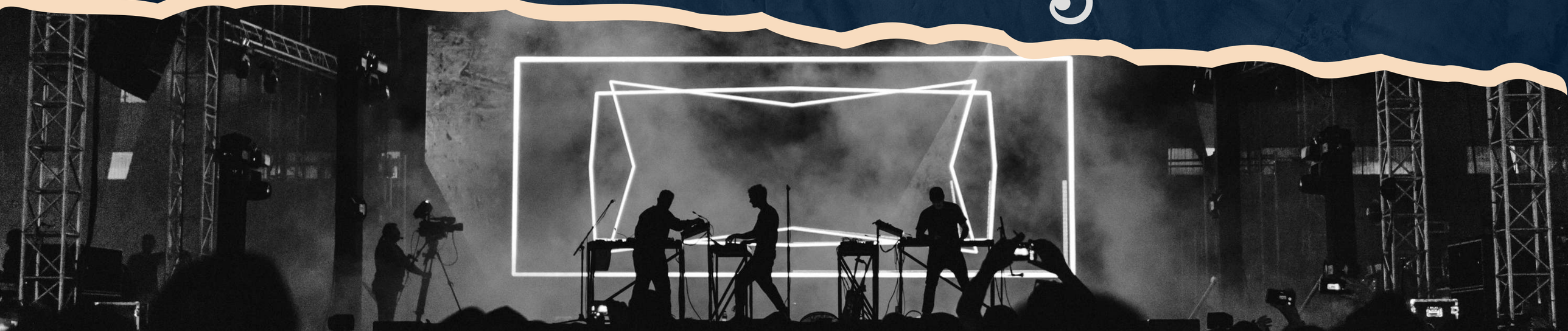
SQL PROJECT

MUSIC STORE ANALYSIS

Amit Gaud

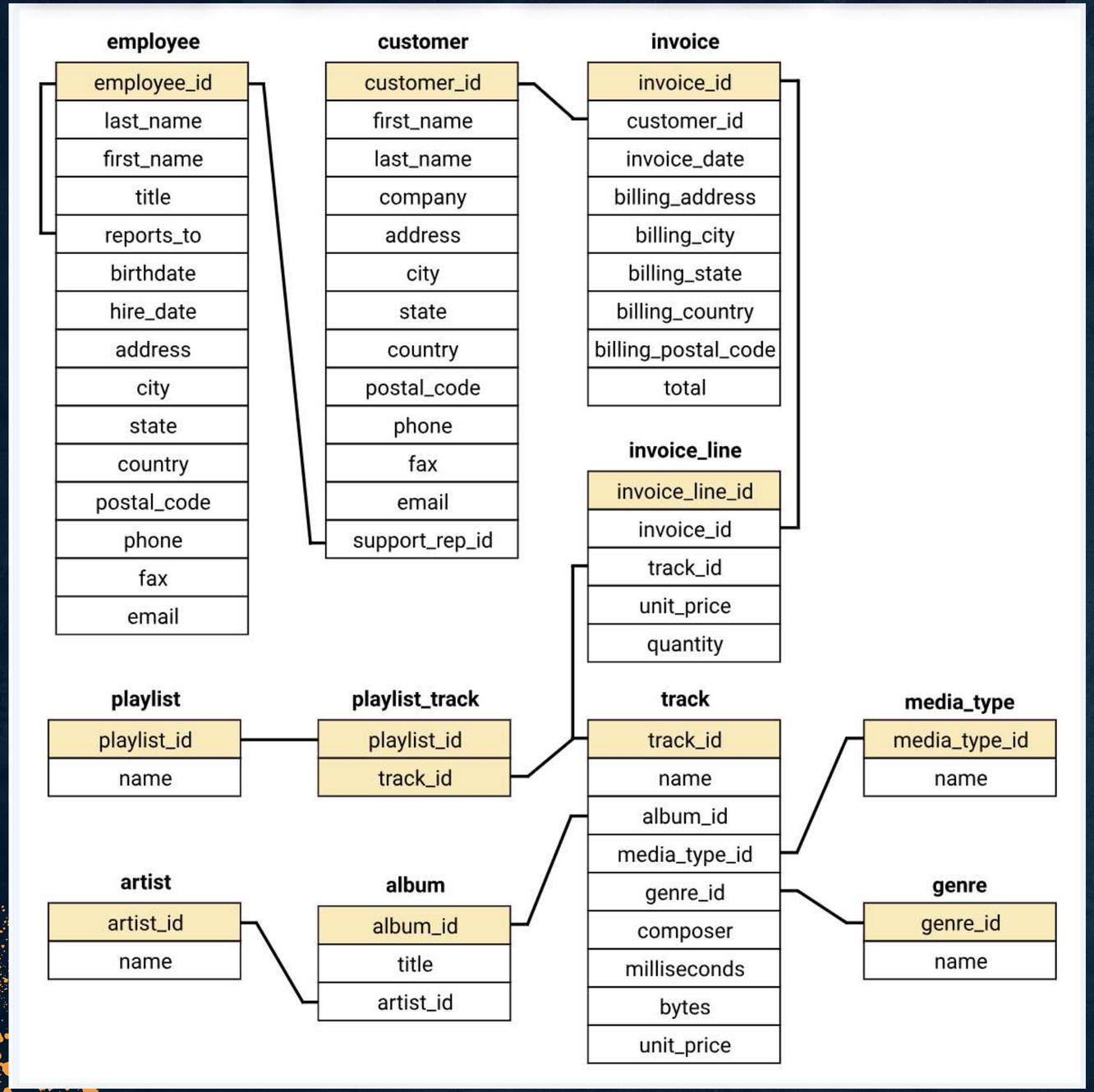


OBJECTIVE

- 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



DIVISION OF QUESTIONS

Queries include :

SELECT, GROUP BY, ORDER BY , LIMIT, DESC

Queries include :

JOINS, GROUP BY , ORDER BY , LIMIT

Queries include :

CTE (COMMON TABLE EXPRESSIONS)

EASY QUESTION- 1

Who is the senior most employee based on job title?

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

	title	first_name	last_name	levels
▶	General Manager	Andrew	Adams	L6

QUESTION- 2

Which countries have the most Invoices?

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

Country	Invoices
USA	131
Canada	76
Brazil	61
France	50
Germany	41
Czech Republic	30
Portugal	29
United Kingdom	28
India	21
Ireland	13
Chile	13
Finland	11
Spain	11
Poland	10

QUESTION- 3

What are top 3 values of total invoice?

```
SELECT ROUND(total,2) AS Total_Invoice
FROM invoice
ORDER BY total DESC
LIMIT 3;
```

Total_Invoice
23.76
19.8
19.8

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.

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

billing_city	invoice_total
Prague	273.24

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.

```
SELECT customer.customer_id, first_name, last_name, ROUND(SUM(total),2) AS Total_Spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id, first_name, last_name
ORDER BY Total_spending DESC
LIMIT 1;
```

customer_id	first_name	last_name	Total_Spending
5	FrantiÅiek	WichterlovÃi	144.54

MODERATE

QUESTION- 1

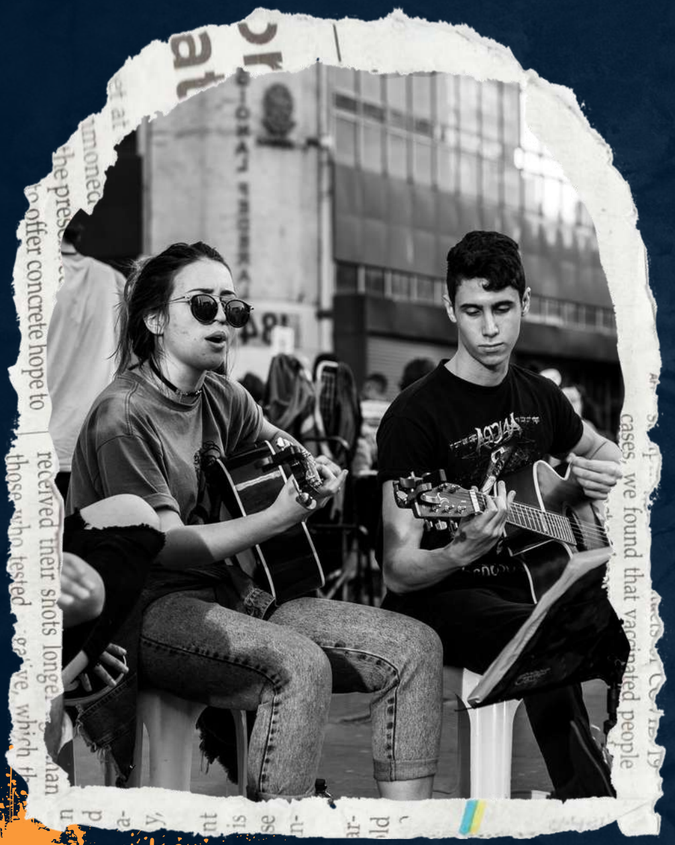
Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A.

```
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;
```

Email	FirstName	LastName	Name
aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
alero@uol.com.br	Alexandre	Rocha	Rock
astrid.gruber@apple.at	Astrid	Gruber	Rock
bjorn.hansen@yahoo.no	BjÃ¸rn	Hansen	Rock
camille.bernard@yahoo.fr	Camille	Bernard	Rock
daan_peeters@apple.be	Daan	Peeters	Rock
diego.gutierrez@yahoo.ar	Diego	GutiÃ©rrez	Rock
dmiller@comcast.com	Dan	Miller	Rock

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.



```
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;
```

Artist_Name	Total_track
AC/DC	18
Aerosmith	15
Audioslave	14
Led Zeppelin	14
Alanis Morissette	13
Alice In Chains	12
Frank Zappa & Captain Beefheart	9
Accept	4

QUESTION- 3

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.

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

Track_Name	milliseconds
How Many More Times	711836
Advance Romance	677694
Sleeping Village	644571
You Shook Me(2)	619467
Talkin' 'Bout Women Obviously	589531
Stratus	582086
No More Tears	555075
The Alchemist	509413

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

```
WITH best_selling_artist AS (
    SELECT artist.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 1,2
    ORDER BY 3 DESC
    LIMIT 1
)
SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
ROUND(SUM(il.unit_price*il.quantity),2) AS Total_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;
```

OUTPUT

customer_id	first_name	last_name	artist_name	Total_Spent
54	Steve	Murray	AC/DC	17.82
53	Phil	Hughes	AC/DC	10.89
21	Kathy	Chase	AC/DC	10.89
49	StanisÅ‚aw	WÃ³jcik	AC/DC	9.9
1	LuÃ-s	GonÃ§alves	AC/DC	7.92
24	Frank	Ralston	AC/DC	7.92
31	Martha	Silk	AC/DC	3.96
16	Frank	Harris	AC/DC	2.97
42	Wyatt	Girard	AC/DC	2.97
6	Helena	HolÃ½	AC/DC	2.97
38	Niklas	SchrÃ¶der	AC/DC	2.97
35	Madalena	Sampaio	AC/DC	2.97
44	Terhi	HÃ¤mÃ¤lÃ…	AC/DC	2.97
9	Kara	Nielsen	AC/DC	1.98
34	JoÃ£o	Fernandes	AC/DC	1.98
57	Luis	Rojas	AC/DC	1.98

QUESTION- 2

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 most_popular_genre AS(
SELECT customer.country AS Country, COUNT(invoice_line.quantity) AS Purchases,
genre.genre_id AS Genre_ID, genre.name AS Genre_Name,
ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) as Row_No
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
GROUP BY Country, genre.genre_id, genre.name
ORDER BY Country, Purchases DESC
)
SELECT Country, Purchases, Genre_ID, Genre_Name
FROM most_popular_genre WHERE Row_No <=1;
```

OUTPUT

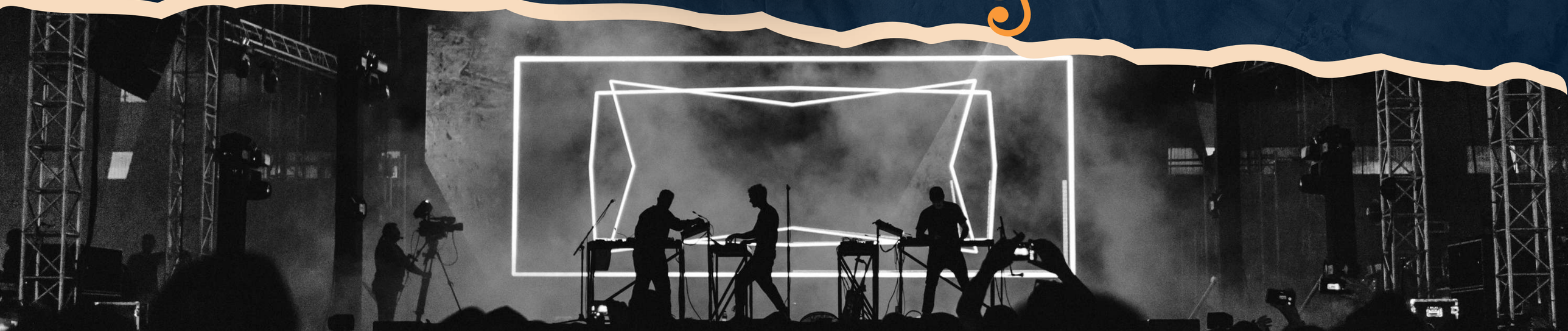
Country	Purchases	Genre_ID	Genre_Name
Argentina	1	1	Rock
Australia	18	1	Rock
Austria	6	1	Rock
Belgium	5	1	Rock
Brazil	26	1	Rock
Canada	57	1	Rock
Chile	7	1	Rock
Czech Republic	14	1	Rock
Denmark	6	1	Rock
Finland	6	1	Rock
France	26	1	Rock
Germany	28	1	Rock
Hungary	4	1	Rock
India	13	1	Rock
Ireland	2	1	Rock
Italy	3	1	Rock

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.

```sql
WITH Customter_with_country AS
(
SELECT customer.customer_id, customer.first_name, customer.last_name,
invoice.billing_country AS Country, ROUND(SUM(invoice.total),2) AS total_spent,
ROW_NUMBER() OVER(PARTITION BY invoice.billing_country ORDER BY SUM(invoice.total) DESC) AS Row_No
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
GROUP BY 1,2,3,4
ORDER BY 4,5 DESC
)
SELECT customer_id, first_name, last_name, Country, total_spent
FROM Customter_with_country
WHERE Row_No <= 1;
```

OUTPUT

customer_id	first_name	last_name	Country	total_spent
56	Diego	GutiÃ©rrez	Argentina	39.6
55	Mark	Taylor	Australia	81.18
7	Astrid	Gruber	Austria	69.3
8	Daan	Peeters	Belgium	60.39
1	LuÃ-s	GonÃ§alves	Brazil	108.9
3	FranÃ§ois	Tremblay	Canada	99.99
57	Luis	Rojas	Chile	97.02
5	FrantiÅiek	WichterlovÃi	Czech Republic	144.54
9	Kara	Nielsen	Denmark	37.62
44	Terhi	HÃ¤mÃ¤lÃ¤inen	Finland	79.2
42	Wyatt	Girard	France	99.99
37	Fynn	Zimmermann	Germany	94.05
45	Ladislav	KovÃics	Hungary	78.21
58	Manoj	Pareek	India	111.87
46	Hugh	O'Reilly	Ireland	114.84
47	Lucas	Mancini	Italy	50.49
48	Johannes	Van der Berg	Netherlands	65.34
4	BjÃ¸rn	Hansen	Norway	72.27
49	StanisÅ‚aw	WÃ³jcik	Poland	76.23

THANK YOU