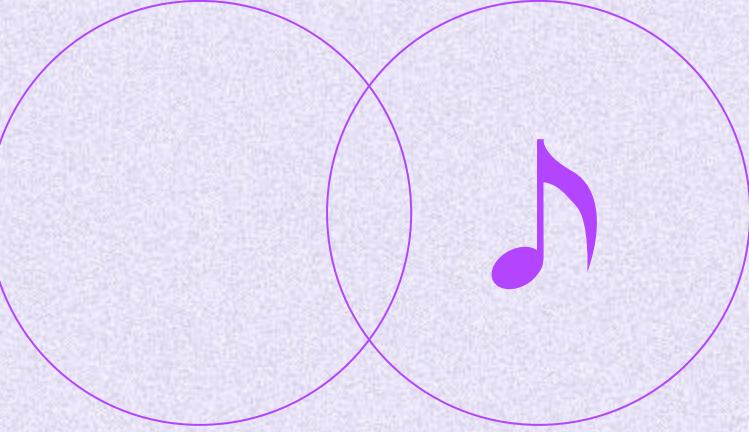


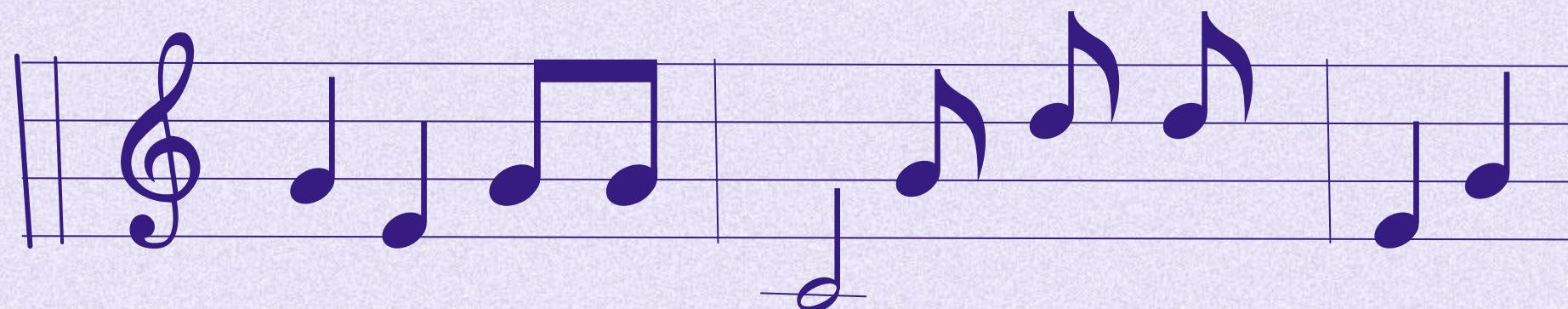


SQL PROJECT



Music Store Analysis

-Sanskriti Chavan.





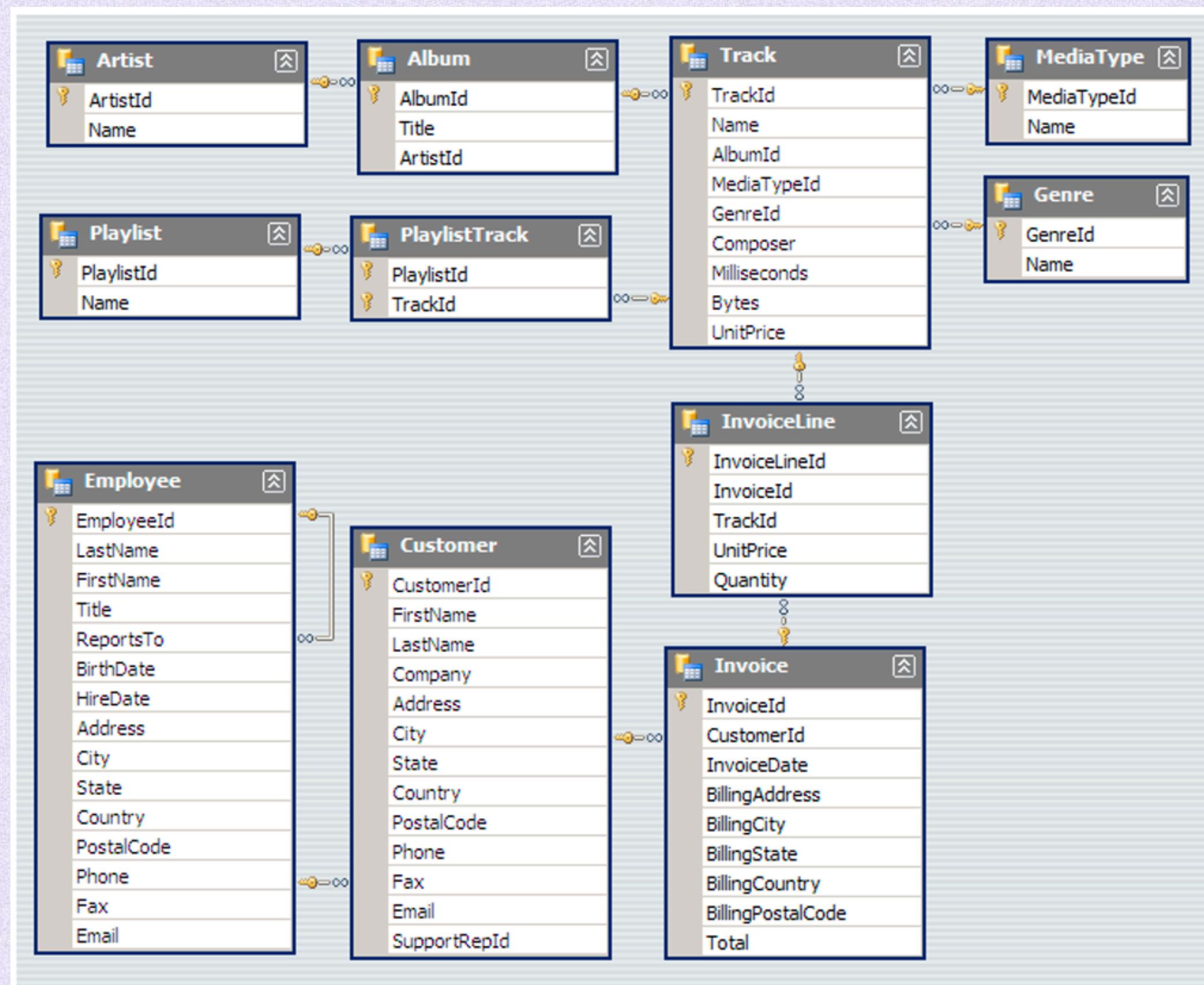
OBJECTIVE



- ♣ The primary objective of The Music Store is to achieve sustainable business growth while addressing existing challenges.
- ♣ We need to explain the dataset with SQL and help the music store understand its business growth by answering simple questions.

Music Store Analysis

Music Playlist Database Schema





Division Of Questions



Easy

Queries include :

SELECT, ORDER BY, GROUP
BY, DESC, LIMIT



Moderate

Queries include :

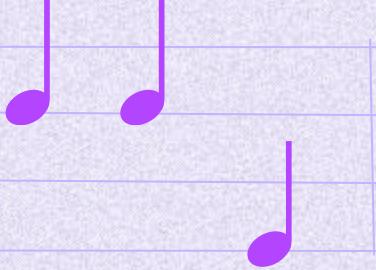
JOINS, ORDER BY, GROUP
BY, DESC



Advanced

Queries include :

CTE(Common Table
Expression)



Question - 1

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

- **SELECT**

```
employee_id, title, first_name, last_name, levels
```

```
FROM
```

```
employee
```

```
ORDER BY levels DESC
```

```
LIMIT 1;
```

A screenshot of a database query results grid. The grid has a header row with columns labeled: employee_id, title, first_name, last_name, and levels. Below the header, there is one data row. The data row contains the values: 1, General Manager, Andrew, Adams, and L6. The grid includes standard database interface elements like 'Result Grid' and 'Filter Rows' buttons at the top.

| | employee_id | title | first_name | last_name | levels |
|---|-------------|-----------------|------------|-----------|--------|
| ▶ | 1 | General Manager | Andrew | Adams | L6 |



Question - 2

♪ Which countries have the most Invoices?

- **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 |
| | Denmark | 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;
```



Result Grid | Filter Rows:

| | Total_Invoice |
|---|---------------|
| ▶ | 23.76 |
| | 19.8 |
| | 19.8 |



Question - 4

♪ Which city has the best customers? We would like to throw 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;
```

The screenshot shows a database query results window. At the top, there are buttons for "Result Grid" (highlighted in orange), "Filter Rows:", and a refresh icon. Below the buttons is a table with two columns: "billing_city" and "invoice_total". A single row is displayed, showing "Prague" in the first column and "273.24" in the second column. The table has a header row and a data row.

| | billing_city | invoice_total |
|---|--------------|---------------|
| ▶ | Prague | 273.24 |

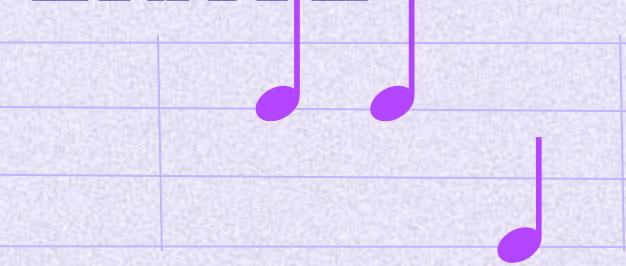


Question - 5

♪ 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 | František | Wichterlová | 144.54 |

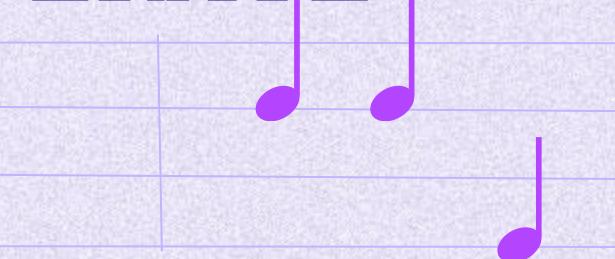


Question - 1

♪ 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 First_Name, last_name AS Last_Name, 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 | First_Name | Last_Name | Name |
|--|-------------------------------|------------|-----------|------|
| | rishabh_mishra@yahoo.in | Rishabh | Mishra | Rock |
| | robbrown@shaw.ca | Robert | Brown | Rock |
| | roberto.almeida@riotur.gov.br | Roberto | Almeida | Rock |
| | stanisÅaw.wÅ³jcik@wp.pl | StanisÅaw | WÅ³jcik | Rock |
| | steve.murray@yahoo.uk | Steve | Murray | Rock |
| | terhi.hamalainen@apple.fi | Terhi | HÄmÄlÄ... | Rock |
| | tgoyer@apple.com | Tim | Goyer | Rock |

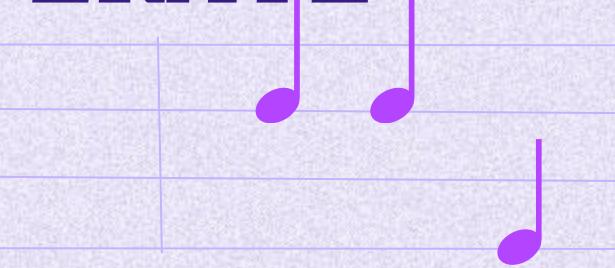


Question - 2

♪ 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 album2 ON album2.artist_id = artist.artist_id
 JOIN track ON track.album_id = album2.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 Beefhe... | 9           |
|   | Accept                          | 4           |

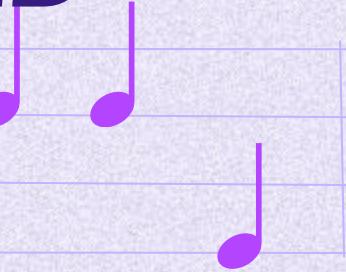


## Question - 3

♪ 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       |
|   | Wheels Of Confusion / The Stra... | 494524       |
|   | Book Of Thel                      | 494393       |
|   | You Oughta Know (Alternate)       | 491885       |
|   | Terra                             | 482429       |
|   | Snoopy's search-Red baron         | 456071       |
|   | Sozinho (Hitmakers Classic Mix)   | 436636       |



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

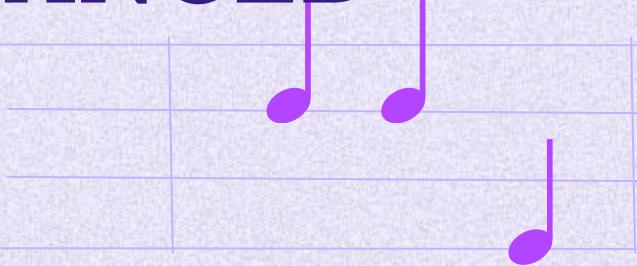
## Question - 1

```
• 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 album2 ON album2.album_id = track.album_id
 JOIN artist ON artist.artist_id = album2.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 album2 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          | StanisÅaw  | 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        |
|   | 27          | Patrick    | Gray       | AC/DC       | 1.98        |
|   | 20          | Dan        | Miller     | 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 = invoice.invoice_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;
```

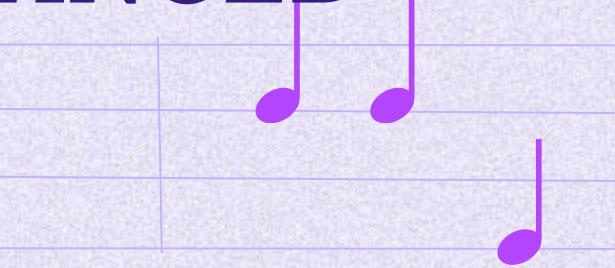
# Music Store Analysis

ADVANCED

## Output

Result Grid | Filter Rows:  Export: Wrap Cell Content:

|   | Country        | Purchases | Genre_id | Genre_Name |
|---|----------------|-----------|----------|------------|
| ▶ | Argentina      | 1         | 3        | Metal      |
|   | Australia      | 1         | 3        | Metal      |
|   | Austria        | 1         | 3        | Metal      |
|   | Belgium        | 1         | 3        | Metal      |
|   | Brazil         | 5         | 3        | Metal      |
|   | Canada         | 8         | 3        | Metal      |
|   | Chile          | 1         | 3        | Metal      |
|   | Czech Republic | 2         | 3        | Metal      |
|   | Denmark        | 1         | 3        | Metal      |
|   | Finland        | 1         | 3        | Metal      |
|   | France         | 5         | 3        | Metal      |
|   | Germany        | 4         | 3        | Metal      |
|   | Hungary        | 1         | 3        | Metal      |
|   | India          | 2         | 3        | Metal      |
|   | Ireland        | 1         | 3        | Metal      |
|   | Italy          | 1         | 3        | Metal      |
|   | Netherlands    | 1         | 3        | Metal      |
|   | Norway         | 1         | 3        | Metal      |



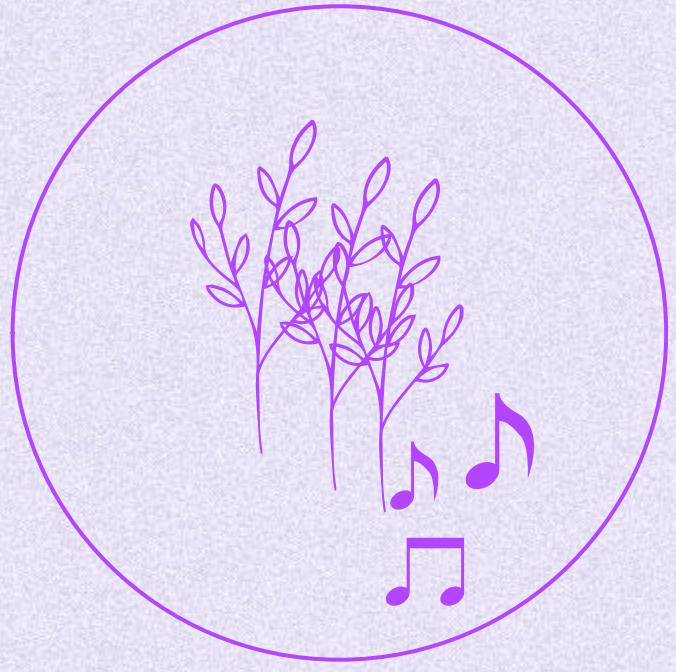
## Question - 3

- ♪ 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.

```
● WITH Customer_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 Customer_with_country
 WHERE Row_No <= 1;
```

## Output

|   | customer_id | first_name | last_name     | Country        | total_spent |
|---|-------------|------------|---------------|----------------|-------------|
| ▶ | 56          | Diego      | Guti  rrez    | Argentina      | 39.6        |
|   | 55          | Mark       | Taylor        | Australia      | 81.18       |
|   | 7           | Astrid     | Gruber        | Austria        | 69.3        |
|   | 8           | Daan       | Peeters       | Belgium        | 60.39       |
|   | 1           | Lu  s      | Gon  alves    | Brazil         | 108.9       |
|   | 3           | Fran  ois  | Tremblay      | Canada         | 99.99       |
|   | 57          | Luis       | Rojas         | Chile          | 97.02       |
|   | 5           | Franti  ek | Wichterlov    | Czech Republic | 144.54      |
|   | 9           | Kara       | Nielsen       | Denmark        | 37.62       |
|   | 44          | Terhi      | H  m  l  inen | Finland        | 79.2        |
|   | 42          | Wyatt      | Girard        | France         | 99.99       |
|   | 37          | Fynn       | Zimmermann    | Germany        | 94.05       |
|   | 45          | Ladislav   | Kov  ics      | Hungary        | 78.21       |
|   | 59          | Manoj      | Dorank        | India          | 111.97      |



Thank  
you!

