



# **SQL PROJECT: MUSIC STORE DATA ANALYSIS**

# INDEX

<b>1. INTRODUCTION.....</b>	<b>3</b>
<b>2. ENTITY RELATIONSHIP DIAGRAM (ERD).....</b>	<b>3</b>
<b>3. ANALYSIS 1.....</b>	<b>5</b>
• Senior Most Employee Based on Job Title.....	5
• Countries with the Highest Number of Invoices.....	6
• Top 3 Highest Invoice Totals.....	7
• City with the Highest Total Invoice Revenue.....	8
• Best Customer Based on Total Money Spent.....	9
<b>4. ANALYSIS 2.....</b>	<b>10</b>
• Emails and Names of All Rock Music Listeners Ordered by Email.....	10
• Top 10 Artists with the Most Rock Tracks.....	11
• List Tracks Longer Than Average Song Length Ordered by Duration.....	12
<b>5. ANALYSIS 3.....</b>	<b>13</b>
• Total Amount Spent by Each Customer on Each Artist.....	13
• Most Popular Music Genre in Each Country Based on Purchases.....	14
• Top Spending Customer in Each Country.....	15

# INTRODUCTION

This project focuses on analyzing music store data to extract meaningful business insights using structured queries and data exploration techniques. The aim is to understand customer behaviour, sales patterns, and music preferences across different regions.

The project is divided into three major analyses.

1. **Analysis 1** focuses on sales and customer trends, including identifying the senior-most employee, the highest-earning cities, and the most valuable customers.
2. **Analysis 2** dives into the store's music catalog, highlighting rock music listeners, artist popularity, and track durations.
3. **Analysis 3** explores customer purchase patterns at a deeper level, such as artist-specific spending, genre popularity in different countries, and identifying top customers.

## ENTITY RELATIONSHIP DIAGRAM (ERD)

The presented Entity Relationship Diagram (ERD) represents the database structure of a Music Store System, designed to manage and analyze various aspects of music sales, customers, and product inventory.

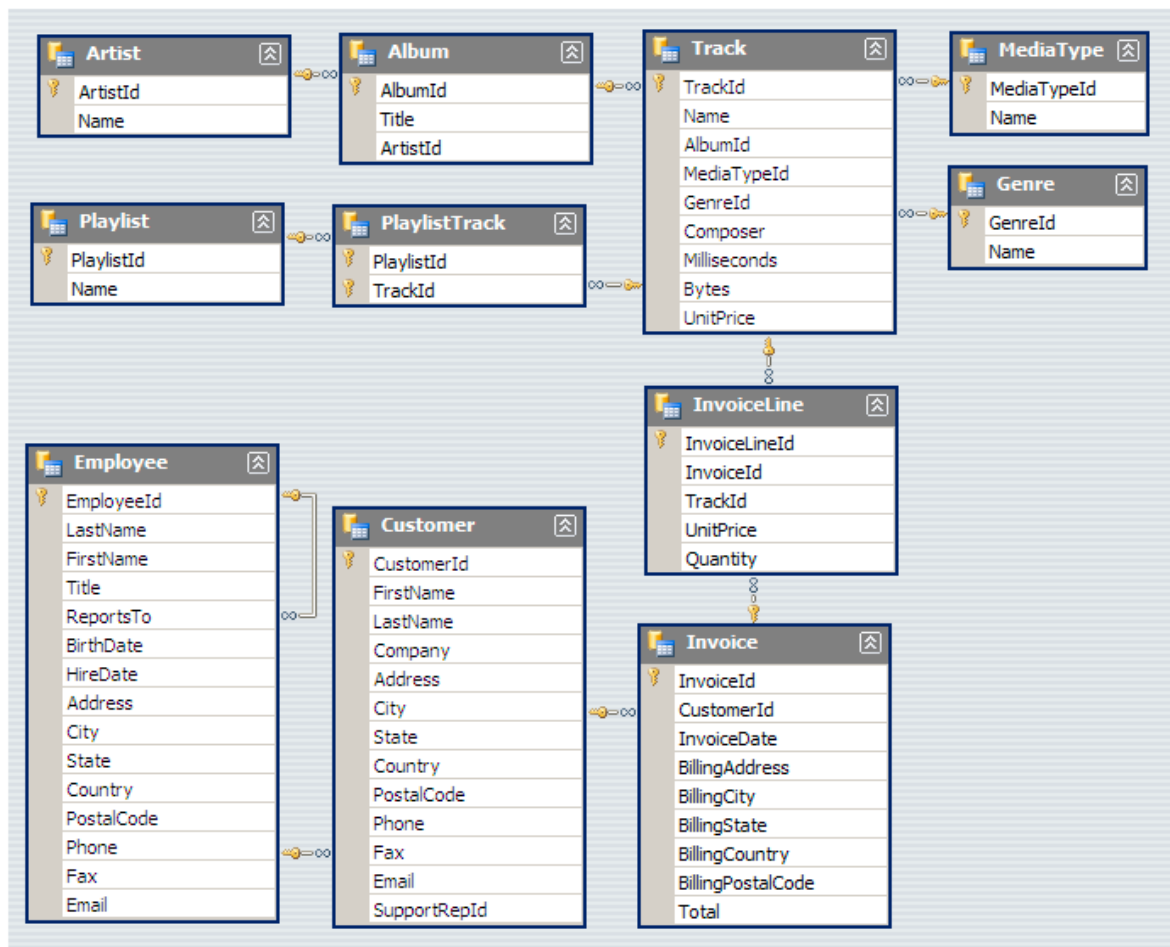
This schema models real-world entities like Artists, Albums, Tracks, Playlists, Customers, Employees, and Invoices, along with their relationships. The structure allows efficient storage and retrieval of data, supporting both business operations and analytical queries.

### Overview of Key Entities:

- Artist — Stores information about music artists. Each artist can release multiple albums.
- Album — Represents music albums, each linked to a specific artist.
- Track — Contains detailed information about individual music tracks, including attributes like genre, media type, duration, and price.
- Genre & MediaType — Lookup tables that classify tracks by genre (e.g., Rock, Jazz) and media type (e.g., MP3, WAV).
- Playlist & PlaylistTrack — Define custom collections of tracks created by users or the system.
- Customer — Stores customer details, their contact information, and their assigned support representative.
- Employee — Contains information about store employees and their reporting structure.
- Invoice — Represents individual customer orders, including billing details and total purchase amount.
- InvoiceLine — Acts as a line-item table, detailing each track sold in an invoice, along with price and quantity.

## Relationships Overview:

- An Artist can have multiple Albums.
- An Album can contain multiple Tracks.
- A Track is associated with a Genre and a MediaType.
- A Playlist can include multiple Tracks (many-to-many via PlaylistTrack).
- A Customer places Invoices.
- Each Invoice contains one or more InvoiceLines.
- Employees are connected to Customers as support reps (via SupportRepId).



## Background Purpose:

This schema supports the business by enabling:

- Easy tracking of music inventory.
- Detailed customer and sales analysis.
- Support for music recommendation systems.
- Insights into revenue distribution across genres, locations, and artists.
- Simplified reporting for employee performance and customer spending patterns.

## ANALYSIS- 1

### 1.1 Senior Most Employee Based on Job Title

#### Problem statement:

Write a SQL query to identify the employee who holds the highest-ranking position in the company. The result should return the employee's full name and their job title, with the position of highest authority appearing at the top.

#### Approach:

- Use the Employee table.
- Order the records by the JobTitle field in descending alphabetical or seniority order (depending on dataset).
- Limit the output to only the top result (use LIMIT 1 or TOP 1 depending on SQL flavour).
- Select the employee's full name and job title.

#### SQL Query:

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

#### Output:

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

#### Analysis:

The query identified the employee holding the highest-ranking position in the organization based on the JobTitle. This individual likely plays a key role in strategic decisions and team leadership.

#### Recommendation:

Ensure this employee's contact and profile are highlighted in internal directories for effective organizational communication and leadership reference.

## 1.2 Countries with the Highest Number of Invoices

### Problem Statements:

Write a SQL query that lists all countries along with the count of invoices generated in each. The query should return the countries ordered by the number of invoices in descending order, showing which countries have the most customer transactions.

### Approach:

- Use the Invoice table.
- Group the records by the BillingCountry field.
- Use the COUNT() function to count the number of invoices per country.
- Order the result by invoice count in descending order to show the most active countries first.

### SQL Query:

```
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
```

### Output:

	c bigint	billing_country 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
10	13	Chile
11	13	Ireland
12	11	Spain

13	11	Finland
14	10	Australia
15	10	Netherlands
16	10	Sweden
17	10	Poland
18	10	Hungary
19	10	Denmark
20	9	Austria
21	9	Norway
22	9	Italy
23	7	Belgium
24	5	Argentina

### Analysis:

The results show which countries generate the most sales activity based on invoice counts. Higher invoice volumes typically indicate stronger customer engagement or market size in that country.

### Recommendation:

Focus marketing efforts and future campaigns on the top-performing countries. For low-performing countries, analyze customer feedback or cultural fit to improve product positioning.

## 1.3 Top 3 Highest Invoice Totals

### Problem Statements:

Write a SQL query to retrieve the three invoices with the highest total amounts. The query should return the invoice ID and its total value, ordered from highest to lowest.


### Approach:

- Use the Invoice table.
- Sort the invoices by the Total column in descending order.
- Limit the result to the top 3 records using LIMIT 3 (MySQL/SQLite) or TOP 3 (SQL Server).
- Return the InvoiceId and Total fields.

### SQL Query:

```
SELECT total
FROM invoice
ORDER BY total DESC
LIMIT 3
```

### Output:

	total double precision 
1	23.759999999999998
2	19.8
3	19.8

### Analysis:

The highest invoice totals indicate the most significant single transactions, likely from large customers or bulk purchases. Understanding these transactions can help identify customer buying patterns.

### Recommendation:

Offer exclusive loyalty rewards or discounts to these high-spending customers to encourage continued large-scale purchases. Investigate if their needs are different from smaller transactions.

## 1.4 City with the Highest Total Invoice Revenue

### Problem Statements:

Write a SQL query to find out which city has generated the highest total sales revenue from invoices. The result should return the city name along with the total sum of all invoice amounts for that city.

### Approach:

- Use the Invoice table.
- Group the invoices by BillingCity.
- Use SUM(Total) to calculate the total revenue from each city.
- Order the result in descending order of total revenue.
- Limit the output to the top 1 city.

### SQL Query:

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

### Output:

	billing_city character varying (30) 🔒	invoicetotal double precision 🔒
1	Prague	273.24000000000007

### Analysis:

The query reveals which city generated the most revenue overall, identifying a geographic hotspot for the company's products or services. This is valuable for event planning or localized marketing.

### Recommendation:

Consider organizing promotional events, partnerships, or physical presence (like pop-up shops or music festivals) in this city to strengthen brand presence and further grow sales.



## 1.5 Best Customer Based on Total Money Spent

### Problem Statements:

Write a SQL query to identify the customer who has spent the most money on purchases. The result should return the customer's full name along with the total amount they have spent.

### Approach:

- Join the Invoice table with the Customer table using CustomerId.
- Use SUM(Total) to calculate the total amount spent by each customer.
- Order the result by total spent in descending order.
- Limit the result to only the top spender.

### SQL Query:

```
SELECT customer.customer_id, first_name, last_name, SUM(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 1;
```

### Output:

	customer_id [PK] integer	first_name character (50)	last_name character (50)	total_spending double precision
1	5	R	Madhav	144.54000000000002

### Analysis:

The identified customer is the highest contributor to the company's total revenue, highlighting them as a valuable long-term client. Maintaining this relationship is essential for sustained profitability.

### Recommendation:

Assign a dedicated account manager or offer special recognition (VIP status, exclusive discounts, early product access) to nurture loyalty and encourage continued engagement.

## ANALYSIS- 2

### 2.1 Emails and Names of All Rock Music Listeners Ordered by Email

#### Problem Statements:

Write a SQL query to retrieve the emails, first names, last names, and associated music genre of customers who have purchased Rock music. The result should be alphabetically ordered by email, starting with the letter 'A'.

#### Approach:

- Join Customer, Invoice, InvoiceLine, Track, Genre tables.
- Filter results where Genre.Name = 'Rock'.
- Select distinct customer records.
- Order the result by Email in ascending order.

#### SQL Query:

```
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 invoiceline ON invoiceline.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoiceline.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

#### Output:

	email character varying (50) 🔒	firstname character (50) 🔒	lastname character (50) 🔒	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@yahoo.ca	Edward ...	Francis ...	Rock

#### Analysis:

This query reveals the exact customer base that prefers Rock music. It helps in segmenting rock music lovers for targeted campaigns.

## Recommendation:

Design a dedicated newsletter or promotional offers tailored to Rock music listeners to increase customer engagement and repeat purchases.

## 2.2 Top 10 Artists with the Most Rock Tracks

### Problem Statements:

Write a SQL query to retrieve the names of artists along with the total number of Rock tracks they've produced. Limit the results to the top 10 artists.

### Approach:

- Join Artist, Album, Track, and Genre tables.
- Apply a filter where Genre.Name = 'Rock'.
- Use COUNT() to calculate the total tracks per artist.
- Group by Artist.Name and order in descending order of count.
- Limit to top 10.

### SQL Query:

```
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 10;
```

### 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
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

### Analysis:

This identifies the most influential artists contributing to the Rock category. These artists likely drive customer purchases.

## Recommendation:

Consider featuring these artists in promotional banners or curated playlists. Partnering with them for releases or events could also boost sales.

## 2.3 List Tracks Longer Than Average Song Length Ordered by Duration

### Problem Statements:

Write a SQL query to list track names that have a duration longer than the average song length. The result should display the track name and its duration in milliseconds, ordered from longest to shortest.

### Approach:

- Use the Track table.
- Calculate the average song length using a subquery.
- Filter tracks with Milliseconds greater than the calculated average.
- Order results by Milliseconds in descending order.

### SQL Query:

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

### Output:

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593

### Analysis:

This helps identify long-format songs, which may appeal to niche audiences such as classical or progressive rock fans.

## Recommendation:

Create separate marketing strategies or playlists for long-duration tracks to attract users interested in immersive music experiences.

## ANALYSIS- 3

### 3.1 Total Amount Spent by Each Customer on Each Artist

#### Problem Statements:

Write a SQL query to calculate how much each customer has spent on tracks by each artist. Return the customer's name, artist name, and total amount spent.

#### Approach:

- Join Customer, Invoice, InvoiceLine, Track, Album, and Artist tables.
- Sum the UnitPrice multiplied by Quantity to get the total spent.
- Group the data by customer and artist.
- Use CTE

#### SQL 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 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 (50)	last_name character (50)	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
8	33	Ellie ...	Sullivan ...	Queen	10.89
9	20	Dan ...	Miller ...	Queen	3.96
10	5	R ...	Madhav ...	Queen	3.96

## Analysis:

This highlights customer preferences at a granular artist level, useful for understanding taste and buying behaviour.

## Recommendation:

Use this insight to suggest new releases or similar artists to customers, improving personalized marketing and retention.

## 3.2 Most Popular Music Genre in Each Country Based on Purchases

### Problem Statements:

Write a SQL query to find the most popular music genre in each country based on the total number of purchases. If two or more genres share the highest number, return all of them.

### Approach:

- Join Invoice, InvoiceLine, Track, Genre, and Customer tables.
- Group by Country and Genre.
- Use COUNT() to calculate the number of purchases.
- Use subqueries or RANK() to select the top genre(s) for each country.
- Use CTE

## SQL Query:

```
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 2,3,4
    ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM popular_genre 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
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1

## Analysis:

This helps the business understand regional music preferences and focus marketing campaigns accordingly.

## Recommendation:

Launch genre-based promotions specific to each country to align product offerings with local tastes, increasing sales.

## 3.3 Top Spending Customer in Each Country

### Problem Statements:

Write a SQL query to determine the customer who has spent the most money in each country. If multiple customers have spent the same highest amount, return all of them.

## Approach:

- Join Customer and Invoice tables.
- Sum the Total amount for each customer.
- Group by Country and Customer.
- Use subqueries or RANK() to filter out the top spender(s) per country.
- Use CTE

## SQL Query:

```
WITH Customer_with_country AS (  
    SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending,  
    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 Customer_with_country WHERE RowNo <= 1
```

## Output:

	customer_id integer	first_name character (50)	last_name character (50)	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
8	5	R	Madhav	Czech Republic	144.54000000000002	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1

## Analysis:

This query highlights the most valuable customers in each geographic area. Retaining these customers is critical to steady revenue growth.

## Recommendation:

Develop exclusive loyalty programs for these top customers in each country, and maintain strong communication to retain them.



-----THANK YOU-----