**Project: MusicDB Data Analysis**

**Purpose:**  
This project uses SQL to perform data analysis on a music store database. It answers common business questions such as identifying top customers, best-selling genres, most profitable locations, and top artists.

### **1. Database Structure**

Although the script doesn’t explicitly create tables, it references the following key tables:

| **Table Name** | **Description** |
| --- | --- |
| Employee | Employee details (first/last name, hire date, title, hierarchy level). |
| Customer | Customer contact info (name, email, country). |
| Invoice | Sales transactions with totals, billing city/country, and customer ID. |
| Invoice\_Line | Line items for each invoice (track, quantity, unit price). |
| Track | Song details (title, album, genre, length). |
| Album, Album2 | Albums and their associated artists. |
| Artist | Artist names and IDs. |
| Genre | Music genres. |

### **3. Key Features**

* Uses **CTEs (WITH)** to merge tables and rank results.
* Performs **aggregate analysis** (SUM, COUNT, AVG).
* Leverages **window functions** (RANK(), ROW\_NUMBER()).
* Shows **real-world business insights**:
  + Top customers
  + Genre popularity
  + Artist revenue
  + Market segmentation by country/city

### **4. Possible Improvements**

* Add **indexes** on genre\_id, customer\_id, and invoice\_id for performance.
* Parameterize queries for **dynamic genre or date range selection**.
* Build a **dashboard** (e.g., Power BI, Tableau) that visualizes these results.
* Add **date filters** to support time-series trend analysis

### **2. Queries Breakdown**

#### **Q1: Most Senior Employee**

SELECT CONCAT(first\_name,' ', last\_name) AS EName, hire\_date, title

FROM Employee

ORDER BY levels;

* Orders employees by levels (hierarchy level) to find the top-ranking employee.

#### **Q2: Countries with Most Invoices**

SELECT COUNT(invoice\_id) Total\_Invoice, billing\_country

FROM invoice

GROUP BY billing\_country

ORDER BY Total\_Invoice DESC;

* Groups invoices by billing country.
* Helps identify strongest markets.

#### **Q3: Top 3 Invoices by Value**

SELECT TOP 3 \* FROM invoice

ORDER BY total DESC;

* Shows biggest transactions.

#### **Q4: Best City for Promotional Event**

SELECT TOP 1 SUM(total) Total\_Invoice, billing\_city

FROM invoice

GROUP BY billing\_city

ORDER BY Total\_Invoice DESC;

* Identifies the city with the highest total invoice value.

#### **Q5: Top-Spending Customer**

SELECT TOP 1 C.first\_name + ' ' + C.last\_name AS Full\_Name, SUM(I.total) AS Total\_Spent

FROM Invoice I

INNER JOIN Customer C ON C.customer\_id = I.customer\_id

GROUP BY C.first\_name, C.last\_name

ORDER BY Total\_Spent DESC;

* Finds the single highest-spending customer.

#### **Customers Who Listen to Rock Music**

SELECT DISTINCT CONCAT(C.first\_name,' ', C.last\_name) AS Full\_Name, C.email, G.name AS Genre\_Type

FROM customer C

INNER JOIN invoice I ON C.customer\_id = I.customer\_id

INNER JOIN invoice\_line IL ON I.invoice\_id = IL.invoice\_id

INNER JOIN track T ON IL.track\_id = T.track\_id

INNER JOIN genre G ON T.genre\_id = G.genre\_id

WHERE G.genre\_id = 1;

* Uses genre\_id = 1 (Rock) to filter.
* Lists customer names and emails for marketing.

#### **Top 10 Rock Artists**

WITH CombinedAlbums AS (

SELECT album\_id, artist\_id FROM Album

UNION ALL

SELECT album\_id, artist\_id FROM Album2

)

SELECT TOP 10 A.name AS Artist\_Name, COUNT(\*) AS Total\_Listens

FROM Invoice\_Line IL

INNER JOIN Track T ON IL.track\_id = T.track\_id

INNER JOIN Genre G ON T.genre\_id = G.genre\_id

INNER JOIN CombinedAlbums CA ON T.album\_id = CA.album\_id

INNER JOIN Artist A ON CA.artist\_id = A.artist\_id

WHERE G.name = 'Rock'

GROUP BY A.name

ORDER BY Total\_Listens DESC;

* Merges Album and Album2.
* Counts how many rock tracks each artist sold.
* Ranks top 10.

#### **Tracks Longer than Average**

SELECT DISTINCT name AS Song\_name, milliseconds

FROM track

WHERE milliseconds > (SELECT AVG(milliseconds) FROM track)

ORDER BY milliseconds;

* Finds tracks exceeding the average length.

#### **Customer Spend by Artist**

WITH CombinedAlbums AS (

SELECT album\_id, artist\_id FROM Album

UNION ALL

SELECT album\_id, artist\_id FROM Album2

)

SELECT C.customer\_id, C.first\_name + ' ' +C.last\_name AS Full\_Name, AR.name,

SUM(IL.unit\_price \* IL.quantity) AS Total\_Spent

FROM customer C

INNER JOIN invoice I ON C.customer\_id = I.customer\_id

INNER JOIN invoice\_line IL ON I.invoice\_id = IL.invoice\_id

INNER JOIN track TR ON IL.track\_id = TR.track\_id

INNER JOIN CombinedAlbums CA ON TR.album\_id = CA.album\_id

INNER JOIN artist AR ON CA.artist\_id = AR.artist\_id

GROUP BY C.customer\_id, C.first\_name, C.last\_name, AR.name

ORDER BY Total\_Spent DESC;

* Shows how much each customer spent per artist.

#### **Most Popular Genre per Country**

WITH GenrePopularity AS (

SELECT

C.country,

G.name AS Genre,

COUNT(\*) AS Total\_Purchases,

RANK() OVER (

PARTITION BY C.country

ORDER BY COUNT(\*) DESC

) AS GenreRank

FROM Invoice\_Line IL

INNER JOIN Invoice I ON IL.invoice\_id = I.invoice\_id

INNER JOIN Customer C ON I.customer\_id = C.customer\_id

INNER JOIN Track T ON IL.track\_id = T.track\_id

INNER JOIN Genre G ON T.genre\_id = G.genre\_id

GROUP BY C.country, G.name

)

SELECT country, Genre, Total\_Purchases

FROM GenrePopularity

WHERE GenreRank = 1

ORDER BY Total\_purchases DESC;

* Uses RANK() to find the #1 genre per country.

#### **Top Spender per Country**

WITH CustomerSpending AS (

SELECT

C.customer\_id,

C.first\_name + ' ' + C.last\_name AS Full\_Name,

C.country,

SUM(IL.unit\_price \* IL.quantity) AS Total\_Spent,

ROW\_NUMBER() OVER (

PARTITION BY C.country

ORDER BY SUM(IL.unit\_price \* IL.quantity) DESC

) AS SpendRank

FROM Customer C

INNER JOIN Invoice I ON C.customer\_id = I.customer\_id

INNER JOIN Invoice\_Line IL ON I.invoice\_id = IL.invoice\_id

GROUP BY C.customer\_id, C.first\_name, C.last\_name, C.country

)

SELECT customer\_id, Full\_Name, country, Total\_Spent

FROM CustomerSpending

WHERE SpendRank = 1

ORDER BY country;

* Finds the highest spender in each country.