SQL Project - Music Store Data Analysis

- This document presents a detailed analysis of a music store dataset using T-SQL(SQL Server). The analysis aims to provide valuable insights into various aspects of the business, such as employee tenure, sales performance across different regions, revenue distribution, customer spending behavior, and opportunities for strategic decision-making and business growth.

Introduction to the Dataset

- The music store dataset comprises a collection of tables that provide comprehensive information about the business's operations. These tables include details on employee, customer, invoice, invoice_line, track, album, artist, playlist_track, playlist, media_type and genre. By analyzing this dataset, we can gain valuable insights into customer demographics, product popularity, sales performance, and other key business metrics.

Questions and Queries

- This section outlines a series of questions that we seek to answer through our data analysis. Each question is accompanied by the corresponding SQL query used to retrieve the necessary data. The queries are designed to extract specific information related to customer behavior, sales trends, product performance, and other key aspects of the music store's business. By examining the results of these queries, we can gain a deeper understanding of the data and identify valuable insights.

Question Set 1 – Easy

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

   Query: Identify the employee with the earliest hire date for each job title.

   ```
   SELECT TOP(1) Title, CONCAT(Fname, ' ', Lname) AS [Full Name]
   FROM Employee
   ORDER BY Levels DESC
   ```

   Insights: Understand the experience levels within the organization.

2. **Which countries have the most Invoices?**

   Query: Determine countries with the highest number of invoices.

   ```
   SELECT Billing_country, COUNT(Customer_id) AS [Number_of_Customers]
   FROM Invoice
   GROUP BY Billing_country
   ORDER BY [Number_of_Customers] DESC
   ```

   Insights: Identify top performing markets for the music store.

3. **What are top 3 values of total invoice?**

   Query: Find the top 3 invoice values.

   ```
   SELECT TOP(3)Total
   FROM Invoice
   ORDER BY Total DESC
   ```

   Insights: Highlight the highest revenue-generating transactions.

4. **Which city has the best customers? We would like to throw a promotional Music Festival in the city where 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.**

   Query: Identify the city with the highest sum of invoice totals.

   ```
   SELECT TOP(1)Billing_city, SUM(Total) AS [Invoice_Total]
   FROM Invoice
   GROUP BY Billing_city
   ORDER BY [Invoice_Total] DESC
   ```

   Insights: Target city for promotional events based on customer spending.

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.**

Query: Determine the customer who has spent the most money.

```
SELECT TOP(1)C.Customer_id, CONCAT(C.Fname, '
',C.Lname) AS [Best_Customer], SUM(Total) AS
[Total_Spending]
FROM Customer C INNER JOIN Invoice I
ON C.Customer_id = I.Customer_id
GROUP BY C.Customer_id, CONCAT(C.Fname, ' ',C.Lname)
ORDER BY [Total_Spending] DESC
```

Insights: Recognize high-value customers for tailored strategies.

Question Set 2 – Intermediate

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.**

   Query: Retrieve email, first name, last name, & genre of Rock Music listeners.

```
SELECT DISTINCT C.Email, C.Fname, C.Lname,
G.Genre_name
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 T.Track_id = IL.Track_id
        INNER JOIN Genre G
  ON G.Genre_id = T.Genre_id AND G.Genre_name = 'Rock'
ORDER BY C.Email ASC
```

   Insights: Understand the Rock Music listener demographics

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.**

Query: Identify the top 10 rock bands with the highest track count.

```
SELECT TOP(10)A.Artist_id, A.Artist_name,
COUNT(A.Artist_id) AS Number_of_Tracks
FROM Track T INNER JOIN Album Alb
        ON Alb.Album_id = T.Album_id
        INNER JOIN Artist A
        ON A.Artist_id = Alb.Artist_id
        INNER JOIN Genre G
ON G.Genre_id = T.Genre_id AND G.Genre_name = 'Rock'
GROUP BY A.Artist_id, A.Artist_name
ORDER BY Number_of_Tracks DESC
```

Insights: Recognize prolific rock music artists.

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 length with the longest songs listed first.**

Query: Retrieve track names with song length longer than average.

```sql
SELECT Track_name, MilliSeconds
FROM Track
WHERE MilliSeconds > (SELECT AVG(MilliSeconds) AS
Avg_Track_Length FROM Track)
ORDER BY MilliSeconds DESC
```

Insights: Highlight tracks with extended durations.

Question Set 3 – Intermediate

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

   Query: Calculate total amount spent by each customer on artists.

```
SELECT C.Customer_id, C.Fname, A.Artist_name,
SUM(IL.Unit_price*IL.Quantity) AS [Total_Spending]
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 T.Track_id = IL.Track_id
    INNER JOIN Album Alb
    ON Alb.Album_id = T.Album_id
    INNER JOIN Artist A
    ON A.Artist_id = ALB.Artist_id
GROUP BY C.Customer_id, C.Fname, A.Artist_name
ORDER BY [Total_Spending] DESC
```

   Insights: Analyze customer preferences and spending habits.

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.**

   Query: Determine the most popular music genre for each country.

```
SELECT *
FROM ( SELECT C.Country, COUNT(IL.Quantity) AS
[Total_Sales], G.Genre_id, G.Genre_name,
    ROW_NUMBER() OVER(PARTITION BY C.Country
ORDER BY COUNT(IL.Quantity) DESC) AS RN
    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 T.Track_id = IL.Track_id
    INNER JOIN Genre G
    ON G.Genre_id = T.Genre_id
GROUP BY C.Country, G.Genre_id, G.Genre_name) AS Temp
WHERE RN = 1
```

   Insights: Identify genre preferences across different regions.

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 customers and how much they spend. For countries where the top amount spent is shared, provide all customers who spent this amount.**

Query: Identify the top customer spending for each country.

```
SELECT *
FROM ( SELECT C.Customer_id, C.Fname, C.Lname,
I.Billing_country, SUM(I.Total) AS [Total_Spending],
     ROW_NUMBER() OVER(PARTITION BY I.Billing_country
ORDER BY SUM(I.Total) DESC) AS RN
     FROM Customer C INNER JOIN Invoice I
     ON C.Customer_id = I.Customer_id
     GROUP BY C.Customer_id,C.Fname, C.Lname,
I.Billing_country) AS Temp
WHERE RN = 1
```

Insights: Recognize high-spending customers in different countries.