# SQL END TO END PROJECT FOR MUSIC STORE ANALYSIS

## Description

This project focuses on utilizing SQL to analyze sales and customer data from an online music store. A real-world situation where a business seeks to analyze consumer behavior and buying patterns in order to make better business decisions is reflected in the dataset.   
SQL Server Management Studio (SSMS) was used for the entire analysis, which followed clear, effective query writing guidelines. In addition to showcasing useful SQL skills, this project also shows how to apply sophisticated analytical functions to solve actual business challenges.

## Tables used in this project

* Artist
* Album
* Track
* Media\_Type
* Genre
* Playlist
* Playlist\_Track
* Customer
* Invoice
* Invoice\_Line
* Employee

## Technologies

## 

* **Database** : SQL Server Management Studio (SSMS)
* **Language** : Structured Query Language (SQL)

## Results

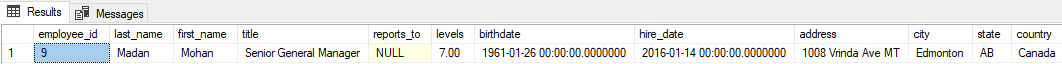
**Question Set 1 - Easy**

Q1: Who is the senior most employee based on job title?

Query :

select top (1) \* from employee order by levels desc;

Result :



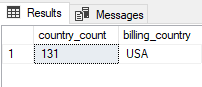
Q2: Which countries have the most Invoices?

Query :

select top (1) count(\*) as country\_count, billing\_country from invoice

group by billing\_country order by country\_count desc;

Result :

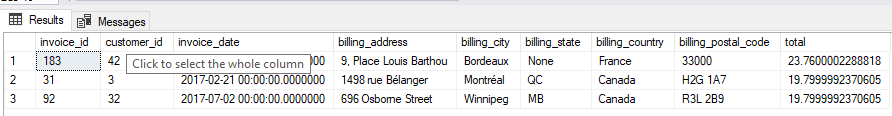


Q3: What are top 3 values of total invoice?

Query :

select top(3) \* from invoice order by total desc;

Result :



Q4: 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.

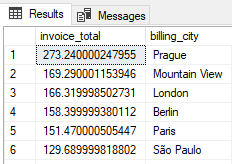
Return both the city name & sum of all invoice totals

Query :

select sum(total) as invoice\_total , billing\_city from invoice

group by billing\_city order by invoice\_total desc;

Result :



Q5: 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 :

select top 1 c.customer\_id, first\_name, last\_name,

sum(invoice.total) as total\_spending

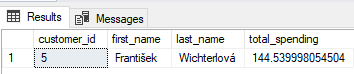
from customer as c

join invoice on c.customer\_id = invoice.customer\_id

group by c.customer\_id, first\_name, last\_name

order by total\_spending desc;

Result :



**Question Set 2 - Moderate**

Q1: 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 :

select distinct c.email , c.first\_name, c.last\_name from customer as c

join invoice as iv on c.customer\_id = iv.customer\_id

join invoice\_line on iv.invoice\_id = invoice\_line.invoice\_id

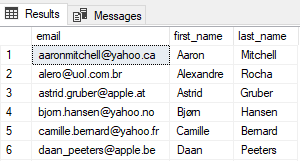
join track on track.track\_id = invoice\_line.track\_id

join genre on genre.genre\_id = track.genre\_id

where genre.name = 'rock'

order by email;

Result :



Q2: 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 :

select top 10

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

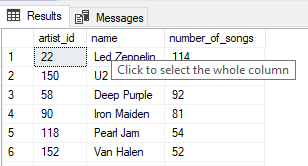
join genre on genre.genre\_id = track.genre\_id

where genre.name like 'Rock'

group by artist.artist\_id, artist.name

order by number\_of\_songs desc;

Result :



Q3: 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.

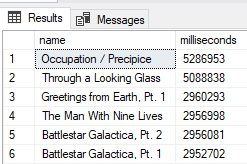
Query :

select distinct name, milliseconds from track where milliseconds >

(select AVG(milliseconds) from track)

order by milliseconds desc;

Result :



**Question Set 3 - Advance**

Q1: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

Query :

with cte as (

select top 1 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 artist.artist\_id, artist.name

order by total\_sales desc

)

select c.customer\_id, c.first\_name, c.last\_name, cte.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

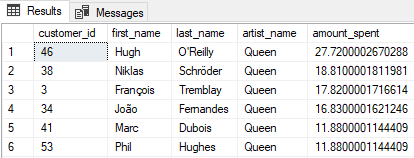
join album alb on alb.album\_id = t.album\_id

join cte on cte.artist\_id = alb.artist\_id

group by c.customer\_id, c.first\_name, c.last\_name, cte.artist\_name

order by amount\_spent desc;

Result :



Q2: 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 :

with cte as (

select g.name , c.country, count(il.quantity) as purchases,

DENSE\_RANK() over (partition by c.country order by count(g.name) desc)

as d\_rank from customer as c

join invoice as i on c.customer\_id = i.customer\_id

join invoice\_line as il on i.invoice\_id = il.invoice\_id

join track as t on il.track\_id = t.track\_id

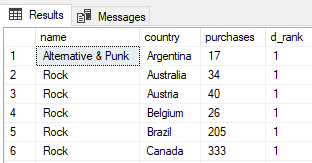
join genre as g on t.genre\_id = g.genre\_id

group by c.country, g.name

)

select \* from cte where d\_rank = 1

Result :



Q3: 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.

Query :

with cte as(

select i.customer\_id , c.first\_name , c.last\_name , billing\_country , sum(total) as total1,

DENSE\_RANK () over (partition by billing\_country order by sum(total) desc) as d\_rank

from invoice as i

join customer as c on i.customer\_id = c.customer\_id

group by i.customer\_id, billing\_country, c.first\_name, c.last\_name

)

select \* from cte where d\_rank = 1;

Result :

