**SQL - Capstone Project:**

## Purposes of The Capstone Project

The major aim of this project is to gain insight into the sales data of Amazon to understand the different factors that affect sales of the different branches.

**About Data:**

This dataset contains sales transactions from three different branches of Amazon, respectively located in Mandalay, Yangon and Naypyitaw. The data contains 17 columns and 1000 rows:

**Analysis List:**

1. **Product Analysis**

Conduct analysis on the data to understand the different product lines, the products lines performing best and the product lines that need to be improved.

1. **Sales Analysis**

This analysis aims to answer the question of the sales trends of product. The result of this can help us measure the effectiveness of each sales strategy the business applies and what modifications are needed to gain more sales.

1. **Customer Analysis**

This analysis aims to uncover the different customer segments, purchase trends and the profitability of each customer segment.

**Approach Used**

1. **Data Wrangling:** This is the first step where inspection of data is done to make sure NULL values and missing values are detected and data replacement methods are used to replace missing or NULL values.

1.1          Build a database

1.2          Create a table and insert the data.

1.3          Select columns with null values in them. There are no null values in our database as in creating the tables, we set NOT  NULL for each field, hence null values are filtered out.

1. **Feature Engineering:** This will help us generate some new columns from existing ones.

2.1           Add a new column named timeofday to give insight of sales in the Morning, Afternoon and Evening. This will help answer the question on which part of the day most sales are made.

2.2          Add a new column named dayname that contains the extracted days of the week on which the given transaction took place (Mon, Tue, Wed, Thur, Fri). This will help answer the question on which week of the day each branch is busiest.

2.3        Add a new column named monthname that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar). Help determine which month of the year has the most sales and profit.

             3. **Exploratory Data Analysis (EDA):** Exploratory data analysis is done to answer the listed questions and aims of this project.

### Business Questions and Answers:

-- 1. What is the count of distinct cities in the dataset ?

Query :

Select Count(Distinct city)

AS distinct\_city\_count

From amazon\_salesdb;

Output:



-- 2. For each branch, what is the corresponding city ?

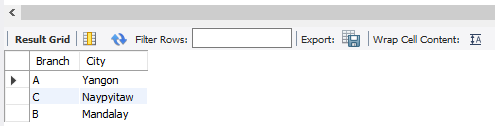
Query :

Select Branch, City

From amazon\_salesdb

Group By Branch, City;

Output :



-- 3. What is the count of distinct product lines in the dataset ?

Query :

Select Count(Distinct product\_line)

AS distinct\_product\_line\_count

From amazon\_salesdb;

Output :



-- 4. Which payment method occurs most frequently ?

Query :

SELECT payment, Count(\*)

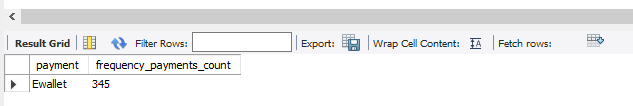
AS frequency\_payments\_count

From amazon\_salesdb

Group by payment

Order by frequency\_payments\_count DESC Limit 1;

Output :



-- 5. Which product line has the highest sales ?

Query :

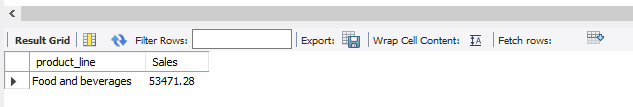
SELECT product\_line, SUM(Quantity \* Unit\_price) AS Sales

FROM amazon\_salesdb

Group By Product\_line

Order By Sales DESC Limit 1;

Output :



-- 6. How much revenue is generated each month ?

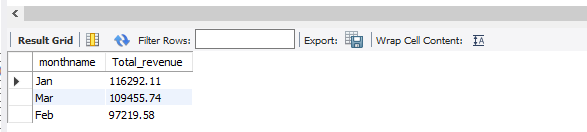
Query :

SELECT monthname, SUM(Total) AS Total\_revenue

FROM amazon\_salesdb

GROUP BY monthname;

Output :



-- 7. In which month did the cost of goods sold reach its peak ?

Query :

SELECT monthname, SUM(COGS) AS cost\_of\_goods\_sold

FROM amazon\_salesdb

GROUP BY monthname

ORDER BY cost\_of\_goods\_sold DESC Limit 1;

Output :



-- 8. Which product line generated the highest revenue ?

Query :

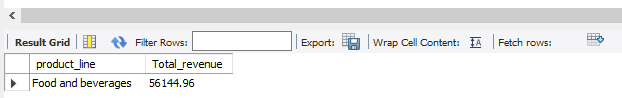
SELECT product\_line, SUM(total) AS Total\_revenue

FROM amazon\_salesdb

GROUP BY product\_line

ORDER BY Total\_revenue DESC Limit 1;

Output :



-- 9. In which city was the highest revenue recorded ?

Query :

SELECT City, SUM(Total) AS Total\_revenue

FROM amazon\_salesdb

GROUP BY City

ORDER BY Total\_revenue DESC Limit 1;

Output :



-- 10. Which product line incurred the highest Value Added Tax ?

Query :

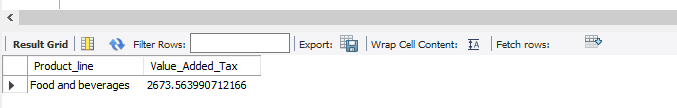
SELECT Product\_line, SUM(Tax\_5\_Percent) AS Value\_Added\_Tax

FROM amazon\_salesdb

GROUP BY Product\_line

ORDER BY Value\_Added\_Tax DESC Limit 1;

Output :



-- 11. For each product line, add a column indicating "Good" if its sales are above

average, otherwise "Bad." ?

Query :

WITH ProductSales AS

(SELECT Product\_line, SUM(Quantity \* Unit\_price) AS Sales

FROM amazon\_salesdb

Group By Product\_line),

AverageSales AS (Select AVG(Sales)

AS Avg\_Sales FROM ProductSales)

SELECT P.Product\_line, P.Sales,

CASE WHEN P.Sales > (SELECT Avg\_Sales FROM AverageSales)

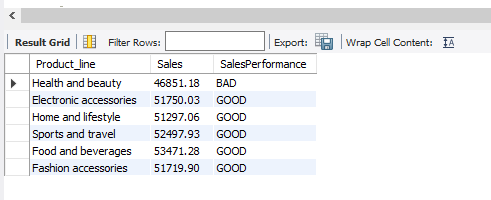
THEN 'GOOD'

ELSE 'BAD'

END AS SalesPerformance

FROM ProductSales P;

Output :



-- 12. Identify the branch that exceeded the average number of products sold ?

Query :

WITH BranchTotalSales AS

(SELECT Branch, SUM(Quantity) AS Total\_Sales

FROM amazon\_salesdb

GROUP BY Branch),

AverageProductsSold AS

(Select AVG(Total\_Sales) AS Avg\_Sales

FROM BranchTotalSales)

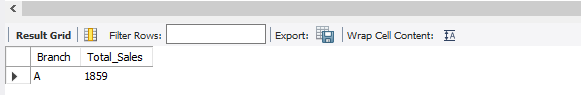
Select B.Branch, B.Total\_Sales

FROM BranchTotalSales B

WHERE B.Total\_Sales > (Select Avg\_Sales

FROM AverageProductsSold);

Output :



-- 13. Which product line is most frequently associated with each gender ?

Query :

WITH ProductLineGender

AS (SELECT Gender, Product\_line, COUNT(Gender) AS count

FROM amazon\_salesdb

GROUP BY Gender, Product\_line),

MaxCountGender

AS (SELECT Gender,MAX(count) AS MaxCount

FROM ProductLineGender

GROUP BY Gender)

SELECT ProductLineGender.Gender,

ProductLineGender.Product\_line, ProductLineGender.count

FROM ProductLineGender

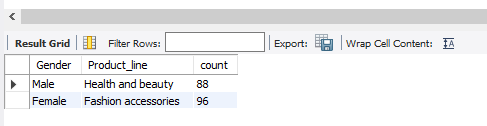
JOIN MaxCountGender ON ProductLineGender.Gender =

MaxCountGender. Gender

AND ProductLineGender.count =

MaxCountGender.MaxCount;

Output :



-- 14. Calculate the average rating for each product line ?

Query :

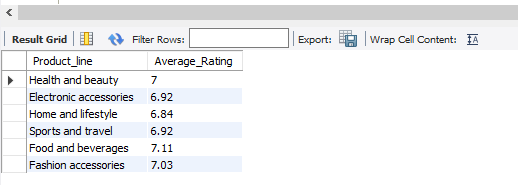
SELECT Product\_line, ROUND(AVG(Rating), 2)

AS Average\_Rating

FROM amazon\_salesdb

GROUP BY Product\_line;

Output :



-- 15. Count the sales occurrences for each time of day on every weekday ?

Query :

SELECT Dayname(date) AS weekday,

CASE

WHEN TIME(time) BETWEEN '06:00:00' AND '11:59:59' THEN 'Morning'

WHEN TIME(time) BETWEEN '12:00:00' AND '17:59:59' THEN 'AFTERNOON'

WHEN TIME(time) BETWEEN '18:00:00' AND '23:59:59' THEN 'Evening'

ELSE 'Night'

END AS time\_of\_day,

COUNT(\*) AS Sales\_Occurrences

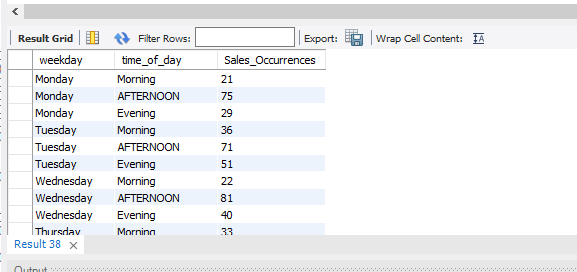
FROM amazon\_salesdb

Group By weekday, time\_of\_day

Order By Field (Weekday, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'),

Field (time\_of\_day, 'Morning', 'Afternoon', 'Evening');

Output :



-- 16. Identify the customer type contributing the highest revenue ?

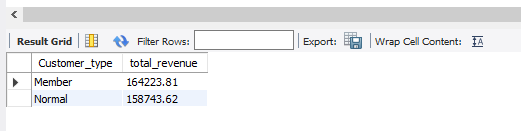
Query :

SELECT Customer\_type, SUM(Total) AS total\_revenue

FROM amazon\_salesdb

GROUP BY Customer\_type;

Output :



-- 17. Determine the city with the highest VAT percentage ?

Query :

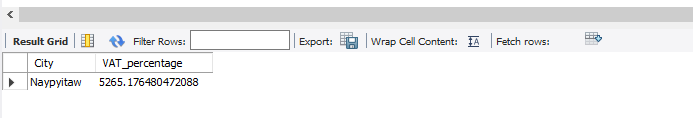
SELECT City, SUM(Tax\_5\_Percent) AS VAT\_percentage

FROM amazon\_salesdb

GROUP BY City

ORDER BY VAT\_percentage DESC Limit 1;

Output :



-- 18. Identify the customer type with the highest VAT payments ?

Query :

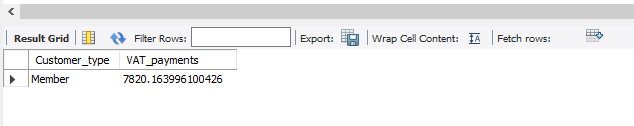
SELECT Customer\_type, SUM(Tax\_5\_percent) AS VAT\_payments

FROM amazon\_salesdb

GROUP BY Customer\_type

Order By VAT\_payments DESC Limit 1;

Output :



-- 19. What is the count of distinct customer types in the dataset ?

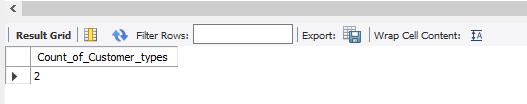
Query :

SELECT COUNT(Distinct Customer\_type)

AS Count\_of\_Customer\_types

FROM amazon\_salesdb;

Output :



-- 20. What is the count of distinct payment methods in the dataset ?

Query :

SELECT COUNT(DISTINCT Payment)

AS count\_of\_payment\_methods

FROM amazon\_salesdb;

Output :



-- 21. Which customer type occurs most frequently ?

Query :

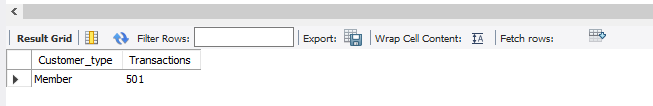
SELECT Customer\_type, COUNT('Invoice\_id') AS Transactions

FROM amazon\_salesdb

GROUP BY Customer\_type

ORDER BY Transactions DESC LIMIT 1;

Output :



-- 22. Identify the customer type with the highest purchase frequency ?

Query :

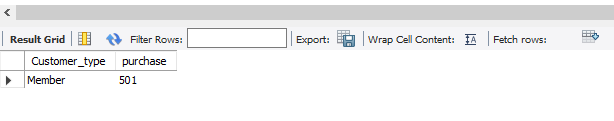
SELECT Customer\_type, COUNT(Quantity) AS purchase

FROM amazon\_salesdb

GROUP BY Customer\_type

ORDER BY purchase DESC LIMIT 1;

Output :



-- 23. Determine the predominant gender among customers ?

Query :

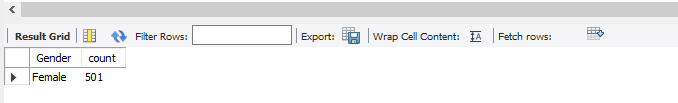
SELECT Gender, COUNT(\*) AS count

FROM amazon\_salesdb

GROUP BY Gender

ORDER BY count DESC LIMIT 1;

Output :



-- 24. Examine the distribution of genders within each branch ?

Query :

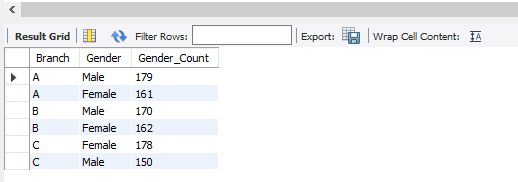
SELECT Branch, Gender, COUNT(\*) AS Gender\_Count

FROM amazon\_salesdb

GROUP BY Branch, Gender

ORDER BY Branch, Gender\_Count DESC;

Output :



-- 25. Identify the time of day when customers provide the most ratings ?

Query :

SELECT TIME(time) AS Rating\_time,

COUNT(\*) AS Rating\_count

FROM amazon\_salesdb

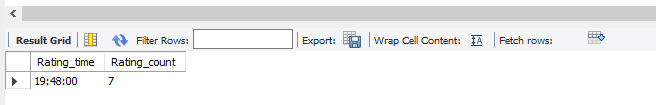
WHERE Rating IS NOT NULL

Group By Rating\_time

Order By Rating\_Count DESC

Limit 1;

Output :



-- 26. Determine the time of day with the highest customer ratings for each branch ?

Query :

WITH Rating AS(

SELECT Branch, timeofday, Avg(Rating)

AS Avg\_Rating

FROM amazon\_salesdb

Group By Branch, timeofday)

SELECT Branch, timeofday, Avg\_Rating

FROM Rating

WHERE (Branch, Avg\_Rating) IN

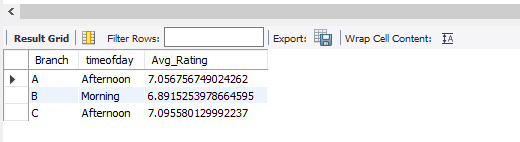
(SELECT Branch, MAX(Avg\_Rating)

FROM Rating

Group By Branch)

Order By Branch;

Output :



-- 27. Identify the day of the week with the highest average ratings ?

Query :

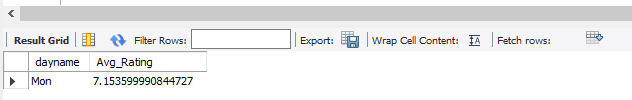
SELECT dayname,AVG(Rating) AS Avg\_Rating

FROM amazon\_salesdb

GROUP BY dayname

ORDER BY Avg\_Rating DESC LIMIT 1;

Output :



-- 28. Determine the day of the week with the highest average ratings for each branch ?

Query :

WITH Rating AS

(SELECT Branch, Dayname, AVG(Rating) AS Avg\_Rating

FROM amazon\_salesdb

GROUP BY Branch, Dayname)

SELECT Branch, Dayname, AVG\_Rating

FROM Rating

WHERE (Branch, Avg\_Rating) IN

(SELECT Branch, MAX(Avg\_Rating)

FROM Rating

GROUP BY Branch)

ORDER BY Branch;

Output :

