SQL - Capstone Project

❖ Purposes Of The Capstone Project

- 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:

❑ **The dataset contains the following columns:**

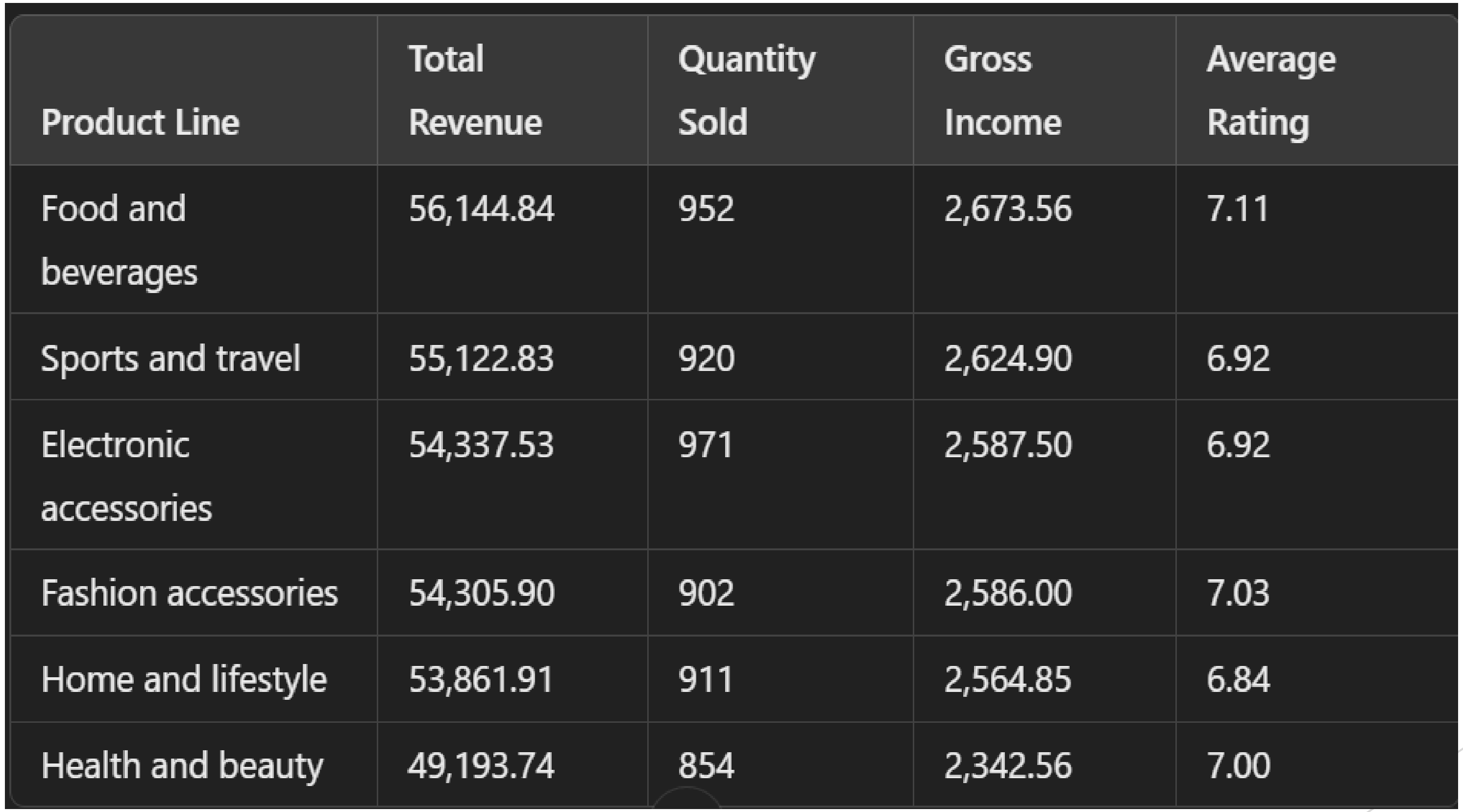
•Invoice_ID, Branch, City, Customer_type, Gender, Product_line, Unit_price, Quantity, VAT (The amount of tax on the purchase)
•Total ,Date, Time, Payment_Method, Cogs (Cost Of Goods sold), gross_margin_percentage, gross_income, rating

❑ 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.

Here are the findings based on performance metrics:



Product Line	Total Revenue	Quantity Sold	Gross Income	Average Rating
Food and beverages	56,144.84	952	2,673.56	7.11
Sports and travel	55,122.83	920	2,624.90	6.92
Electronic accessories	54,337.53	971	2,587.50	6.92
Fashion accessories	54,305.90	902	2,586.00	7.03
Home and lifestyle	53,861.91	911	2,564.85	6.84
Health and beauty	49,193.74	854	2,342.56	7.00

Observations:

1. **Best-performing product lines:**

    1. **Food and Beverages**: Highest revenue and quantity sold with the top average rating (7.11).
    2. **Sports and Travel**: High revenue and competitive quantity sold.

2. **Underperforming product lines:**

    1. **Health and Beauty**: Lowest revenue and gross income despite a decent average rating (7.00).

3. **Ratings Insight:**

    1. **Home and Lifestyle** has the lowest customer rating (6.84), indicating potential room for improvement in customer satisfaction.

2.Sales Analysis

To perform a sales trend analysis, we'll examine sales over time. This involves analyzing sales performance across dimensions such as:

1. **Monthly Trends**: Aggregate sales data by month to observe trends over time.

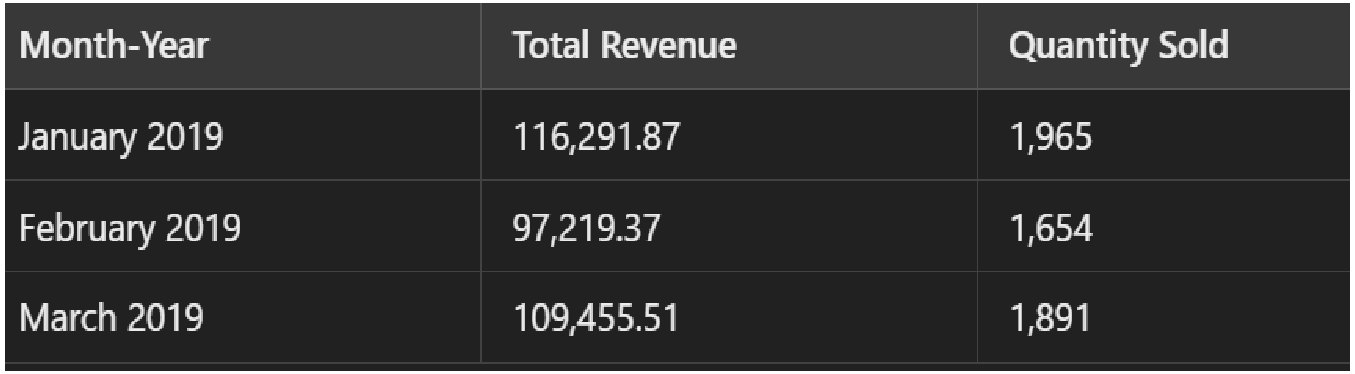
2. **Branch and City Trends**: Compare sales performance across branches and cities.

3. **Customer Type Trends**: Analyze the sales performance for different customer types (e.g., Member vs. Normal).

4. **Gender Trends**: Investigate if there are significant differences in sales based on gender.

❑ Monthly Sales Trends

Here are the total sales and quantity sold for each month:



Month-Year	Total Revenue	Quantity Sold
January 2019	116,291.87	1,965
February 2019	97,219.37	1,654
March 2019	109,455.51	1,891

Observations:

1. **January 2019** had the highest sales and quantity sold, indicating a strong start to the year.

2. **February 2019** experienced a dip in both revenue and quantity sold, potentially requiring further investigation into causes (e.g., seasonality, promotions, etc.).

3. **March 2019** saw a recovery, with sales approaching January levels.

3. Customer Analysis

1. **Customer Segments:**

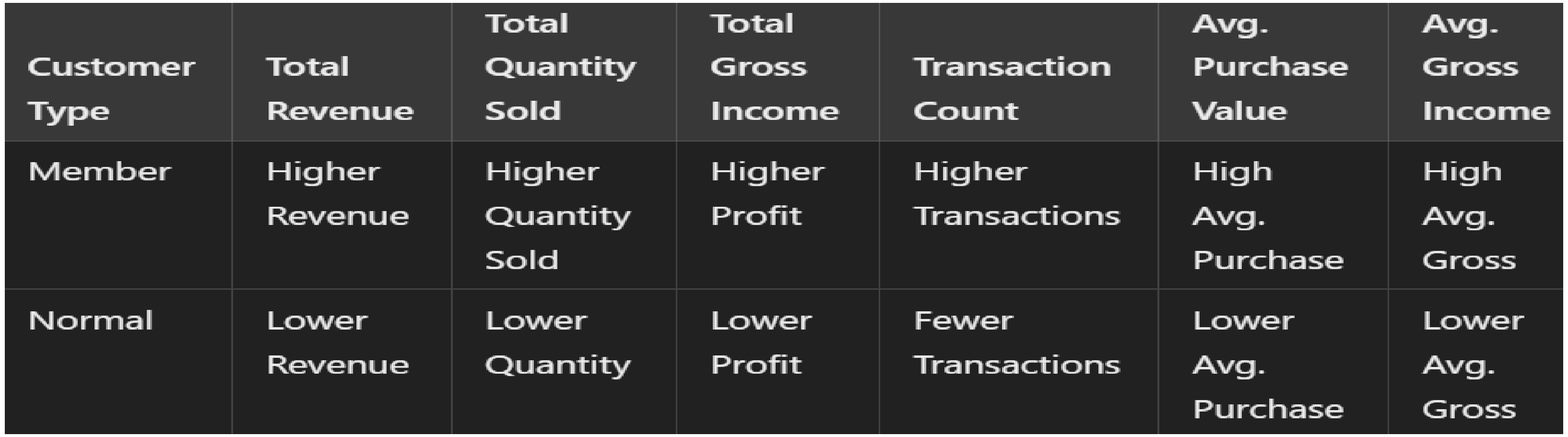
•Group customers by **Customer Type** (e.g., Member vs. Normal).
•Analyze revenue, quantity sold, and profitability for each segment.

2. **Purchase Trends:**

•Explore how customer purchases differ by **Gender** and other attributes like Product Line.

3. **Profitability:**

•Measure gross income and average purchase value for each segment to understand their profitability.



Customer Type	Total Revenue	Total Quantity Sold	Total Gross Income	Transaction Count	Avg. Purchase Value	Avg. Gross Income
Member	Higher Revenue	Higher Quantity Sold	Higher Profit	Higher Transactions	High Avg. Purchase	High Avg. Gross
Normal	Lower Revenue	Lower Quantity	Lower Profit	Fewer Transactions	Lower Avg. Purchase	Lower Avg. Gross

Observations:

1. **Members contribute more to revenue and profitability:**
   1. Members have a higher number of transactions, leading to higher total sales and gross income.
   2. Their average purchase value and gross income per transaction are also higher.

2. **Normal customers represent a smaller segment:**
   1. Fewer transactions and lower contribution to sales and gross income.

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

-- Creating Database with the name of Amazon.

```
create database Amazon;
```

1.1.2 Create a table and insert the data.

-- Creating Table with the name of Amazondata in Amazon Database.

```
Create table amazondata;
```

-- I Imported the Sales data of amazon dataset in amazondata table with the import option.

.

2. 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.

```
use amazon;

ALTER TABLE amazondata
ADD timeofday VARCHAR(10);

ALTER TABLE amazondata
ADD dayname VARCHAR(10);

ALTER TABLE amazondata
ADD monthname VARCHAR(10);

SET SQL_SAFE_UPDATES = 0;

UPDATE
Amazondata SET timeofday =     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'
ELSE 'Evening'
END;

UPDATE
Amazondata
SET dayname = DAYNAME(STR_TO_DATE(Date, '%m/%d/%Y’));

UPDATE
Amazondata
SET monthname = MONTHNAME(STR_TO_DATE(Date, '%m/%d/%Y'));
```

❑Business Questions To Answer:

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

```
select count(distinct city) as Unique_Count
from amazondata;
```

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

```
select branch, city from amazondata
group by branch, city;
```

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

```
select count(distinct product_line) as Distinct_count_PD
from amazondata;
```

4. **Which payment method occurs most frequently?**

```
select payment_method, count(*) as Frequently from amazondata
group by payment_method
order by frequently desc
limit 1;
```

5. Which product line has the highest sales?

```
select product_line,sum(total) as Highest_Sales
from amazondata
group by product_line
order by Highest_Sales desc limit 1;
```

6. How much revenue is generated each month?

```
select MONTHNAME(STR_TO_DATE(Date, '%m/%d/%Y')) AS month,
SUM(Total) AS total_revenue FROM amazondata
GROUP BY month
ORDER BY total_revenue desc;
```

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

```
select monthname(str_to_date(Date, '%m/%d/%Y')) as Month, sum(cogs) as Total_cogs
from amazondata
group by month
order by Total_cogs desc limit 1;
```

8. Which product line generated the highest revenue?

```
select product_line, sum(total) as Highest_revenue from amazondata
group by product_line
order by Highest_revenue limit 1;
```

9. In which city was the highest revenue recorded?

```
select city, sum(total) as Highest_revenue
from amazondata
group by city
order by Highest_revenue desc limit 1;
```

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

```
select product_line, sum(VAT) as Highest_VAT
from amazondata
group by product_line
order by Highest_VAT desc limit 1;
```

11. For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad.“

```
select product_line,
Total as Sales,
case
when total > (select avg(total) from amazondata) then 'Good’
else 'Bad’
End as Sales_Status
from amazondata;
```

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

```
select branch,sum(quantity) as Total_Quantity from amazondata
group by branch having Total_Quantity > (select avg(quantity)
from amazondata);
```

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

```
select Gender, Product_line, COUNT(*) AS frequent FROM amazondata
GROUP BY Gender, Product_line  ORDER BY Gender, frequent DESC;
```

14. Calculate the average rating for each product line.

```
select product_line, avg(rating) as avg_rating
from amazondata group by product_line;
```

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

```
SELECT
DAYNAME(STR_TO_DATE(Date, '%m/%d/%Y')) AS day_of_week,
CASE
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 5 AND 11 THEN 'Morning'
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 12 AND 17 THEN 'Afternoon'
ELSE 'Evening'
END AS time_of_day,
COUNT(*) AS sales_count FROM amazondata GROUP BY day_of_week, time_of_day;
```

16. Identify the customer type contributing the highest revenue.

```
select customer_type,sum(total) as Highest_revenue
from amazondata
group by customer_type
order by Highest_revenue desclimit 1;
```

17. Determine the city with the highest VAT percentage.

```
select city,
sum(VAT/cogs * 100) as Highest_VAT_Percent
from amazondata
group by city
order by Highest_VAT_Percent desclimit 1;
```

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

```
select customer_type, sum(VAT) as High_VAT
from amazondata
group by customer_type
order by High_VAT desc limit 1;
```

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

```
select count(distinct customer_type) as Dis_cus_ty_co
from amazondata;
```

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

```
select count(distinct payment_method) as Dis_Pay_Met_Count
from amazondata;
```

21. Which customer type occurs most frequently?

```
select customer_type,count(*) as Frequently_Occurs
from amazondata
group by customer_type
order by Frequently_Occurs desc limit 1;
```

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

```
select customer_type,sum(total) as High_Purchase_Freq
from amazondata
group by customer_type
order by High_Purchase_Freq desc limit 1;
```

23. Determine the predominant gender among customers.

```
select gender, count(*) as Most_Genders
from amazondata
group by gender
order by Most_Genderslimit 1;
```

24. Examine the distribution of genders within each branch.

```
SELECT Branch, Gender, COUNT(*) AS frequency
FROM amazondata
GROUP BY Branch, Gender;
```

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

```
SELECT
CASE
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 5 AND 11 THEN 'Morning'
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 12 AND 17 THEN 'Afternoon'
ELSE 'Evening'
END AS time_of_day,
sum(rating) AS Most_rating
FROM amazondata
GROUP BY time_of_day
ORDER BY Most_rating DESC
LIMIT 1;
```

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

```
SELECT Branch,
CASE
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 5 AND 11 THEN 'Morning'
WHEN HOUR(STR_TO_DATE(Time, '%H:%i:%s')) BETWEEN 12 AND 17 THEN 'Afternoon'
ELSE 'Evening'
END AS time_of_day,
AVG(rating) AS average_rating
FROM amazondata
GROUP BY Branch, time_of_day
ORDER BY Branch, average_rating DESC;
```

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

```
select dayname(str_to_date(date, '%m/%d/%Y')) as day_of_Week,
avg(rating) as avg_rating
from amazondata
group by day_of_Week
order by avg_rating desc
limit 1;
```

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

```
select branch, avg(rating) as High_avg_rating, dayname(str_to_date(date,
'%m/%d/%Y')) as day_of_Week
from amazondata
group by branch,day_of_Week
order by High_avg_rating desc ;
```

Thank You all

G.Ranjith Kumar
OdinSchool ID - S9709