Amazon Sales Data Analysis using SQL

An End-to-End SQL Portfolio Project

Prepared by:

Sachin

Role:

B.Com Graduate | Aspiring Data Analyst

Tools Used:

SQL (MySQL), MySQL Workbench

Project Type:

Data Analytics Portfolio Project

Year:

2026

Project Overview

This project focuses on analyzing Amazon sales data using SQL to understand sales trends, customer behavior, product performance, and profitability. The analysis is based on transactional sales data collected from multiple Amazon branches during the first quarter of 2019.

The goal of this project is to apply SQL concepts in a real-world business scenario and extract meaningful insights that can support business decisions. By using a combination of basic and advanced SQL techniques such as aggregations, subqueries, CTEs, and window functions, the project answers key business questions related to revenue growth, customer preferences, and operational efficiency.

📊 Dataset Description

The dataset used in this project contains Amazon sales transaction records from three branches located in different cities. Each row represents a single purchase transaction and includes information about the product, customer, payment method, and sales performance.

Dataset Details:

- **Dataset Name:** Amazon Sales Dataset
- **Time Period:** January 2019 – March 2019
- **Total Records:** 1,000
- **Total Columns:** 20
- **Branches:** A, B, C
- **Cities:** Yangon, Mandalay, Naypyitaw

Key Attributes:

- Sales information: quantity, unit price, total sales, VAT, cost of goods sold (COGS)
- Customer information: customer type, gender
- Product details: product line
- Transaction details: date, time, payment method, rating

To improve the analysis, additional time-based columns such as **time of day**, **day name**, and **month name** were created using SQL.

AMAZON SALES REPORT | SUMMARY

1️⃣ Basic Dataset Overview & Descriptive Analysis

Key Findings:

1: Total number of transactions (customer purchases)

```
SELECT
    COUNT(DISTINCT invoice_id) AS customer_count
FROM amazon_sales;
```

customer_count
1000

Explanation :

This query counts the total number of unique purchase transactions using invoice_id. Each invoice represents a completed customer purchase, so this result reflects total transactions rather than unique customers.

2 : customers count by city

```
SELECT
    city, COUNT(*) AS customer_count
FROM
    amazon_sales
GROUP BY city
ORDER BY customer_count DESC;
```

city	customer_count
Yangon	340
Mandalay	332
Naypyitaw	328

Explanation:

This query calculates how many customers made purchases in each city, helping to identify which city has higher customer engagement.

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

```
SELECT
    COUNT(DISTINCT (product_line))
FROM
    amazon_sales;
```

COUNT(DISTINCT (product_line))
6

Explanation:

This query identifies how many unique product categories are available, giving an overview of product variety offered by Amazon.

4: What is the count of distinct customer types in the dataset?

```
SELECT
    COUNT(DISTINCT customer_type) AS distinct_count
FROM
    amazon_sales;
```

distinct_count
2

Explanation:

This query shows the number of different customer segments (such as Member and Normal) present in the dataset.

5: What is the count of distinct payment methods in the dataset?

```
SELECT
    COUNT(DISTINCT payment_method) AS
distinct_payment_method
FROM
    amazon_sales;
```

distinct_payment_method
3

Explanation:

This query determines how many different payment methods customers use, reflecting payment flexibility and preferences.

2 Sales & Revenue Analysis

6: Which product line has the highest sales?

```
SELECT
        product_line,SUM(quantity) AS total_unit_sold
FROM
        amazon_sales
GROUP BY product_line
ORDER BY total_unit_sold DESC
LIMIT 1;
```

product_line	total_unit_sold
Electronic accessories	971

Explanation:

This query identifies the product line with the highest quantity sold, indicating which category has the highest demand.

7: How much revenue is generated each month?

```
SELECT
    month_name, ROUND(SUM(total), 2) AS monthly_revenue
FROM
    amazon_sales
GROUP BY month_name
ORDER BY monthly_revenue DESC;
```

month_name	monthly_revenue
January	116292.11
March	109455.74
February	97219.58

Explanation:

This query calculates total revenue for each month, helping analyze sales trends over time.

8: Which product line generated the highest revenue?

```
SELECT
    product_line, SUM(total) AS total_revenues
FROM
    amazon_sales
GROUP BY product_line
ORDER BY total_revenues DESC;
```

product_line	total_revenues
Food and beverages	56144.96
Sports and travel	55123.00
Electronic accessories	54337.64
Fashion accessories	54306.03
Home and lifestyle	53861.96
Health and beauty	49193.84

Explanation:
This query finds which product line contributes the most to overall revenue, highlighting the most profitable category in terms of sales value.

9: In which city was the highest revenue recorded?

```
SELECT
    city, SUM(total) AS total_revenue
FROM
    amazon_sales
GROUP BY city
ORDER BY total_revenue DESC  LIMIT 1;
```

city	total_revenue
Naypyitaw	110568.86

Explanation:
This query determines which city generates the highest revenue, useful for location-based business decisions.

10: Which payment method generates the highest revenue?

```
SELECT
    payment_method, ROUND(SUM(total), 2) AS total_revenue
FROM amazon_sales
GROUP BY payment_method
ORDER BY total_revenue DESC;
```

payment_method	total_revenue
Cash	112206.76
Ewallet	109993.38
Credit card	100767.29

Explanation:
This query shows which payment method contributes the most revenue, indicating customer payment preferences that drive higher sales.

11: Peak revenue hour

```
SELECT
    HOUR(time) AS hour_of_day, ROUND(SUM(total), 2) AS revenue
FROM amazon_sales
GROUP BY hour_of_day ORDER BY revenue DESC;
```

hour_of_day	revenue
19	39699.58
13	34723.32
10	31421.57
15	31179.57
14	30828.46
11	30377.43
12	26065.94
18	26030.36
16	25226.35
17	24445.29
20	22969.56

Explanation:
This query identifies the hour of the day when revenue is highest, helping businesses optimize staffing and promotions.

12: Revenue contribution % by branch

```
SELECT
    branch,
    ROUND(SUM(total) * 100 / (SELECT
                SUM(total)
            FROM
                amazon_sales), 2 ) AS revenue_pct
FROM
    amazon_sales
GROUP BY branch;
```

branch	revenue_pct
A	32.88
C	34.24
B	32.88

Explanation:
This query calculates the percentage share of total revenue contributed by each branch, allowing comparison of branch performance.

13: Month-over-month revenue growth

```
SELECT
    month_name,
    ROUND(SUM(total), 2) AS revenue,
    ROUND(
        SUM(total) -
        LAG(SUM(total)) OVER (ORDER BY
FIELD(month_name,'January','February','March')),2) AS revenue_change
    FROM  amazon_sales
    GROUP BY month_name;
```

month_name	revenue	revenue_change
January	116292.11	NULL
February	97219.58	-19072.53
March	109455.74	12236.16

Explanation:

This query analyzes how revenue changes from one month to the next, helping identify growth or decline trends.

3 Cost, VAT & Profitability Analysis

14: In which month did the cost of goods sold reach its peak?

```
SELECT
    month_name, ROUND(SUM(cogs), 2) AS cogs_peak
FROM
    amazon_sales
GROUP BY month_name
ORDER BY cogs_peak DESC;
```

month_name	cogs_peak
January	110754.16
March	104243.34
February	92589.88

Explanation:
This query identifies the month with the highest operational cost, useful for cost control and budgeting analysis.

15: Which product line incurred the highest Value Added Tax?

```
SELECT
    product_line, ROUND(SUM(vat), 2) AS high_vat
FROM
    amazon_sales
GROUP BY product_line
ORDER BY high_vat DESC
LIMIT 1 ;
```

product_line	high_vat
Food and beverages	2673.56

Explanation:
This query finds the product line that contributed the most VAT, reflecting higher taxable sales volume.

16: Determine the city with the highest VAT amount.

```
SELECT
    city, MAX(vat) AS vat_amount
FROM
    amazon_sales
GROUP BY city
ORDER BY vat_amount DESC
LIMIT 1;
```

city	vat_amount
Naypyitaw	49.65

Explanation:
This query determines which city records the highest VAT value, indicating higher-value or higher-volume sales

17: Identify the customer type with the highest VAT payments.

```
SELECT
    customer_type, MAX(vat) AS vat_amount
FROM
    amazon_sales
GROUP BY customer_type
ORDER BY vat_amount DESC;
```

customer_type	vat_amount
Member	49.65
Normal	49.49

Explanation:
This query shows which customer type contributes the highest VAT, helping understand which segment generates more taxable revenue.

18: Which product line is the MOST PROFITABLE?

```
SELECT
    product_line, ROUND(SUM(total - cogs), 2) AS total_profit
FROM
    amazon_sales
GROUP BY product_line
ORDER BY total_profit DESC;
```

product_line	total_profit
Food and beverages	2673.68
Sports and travel	2625.07
Electronic accessories	2587.61
Fashion accessories	2586.13
Home and lifestyle	2564.90
Health and beauty	2342.66

Explanation:
This query calculates profit by subtracting COGS from total sales, identifying the product line generating the highest profit.

19: Profit margin (%) by product line

```
SELECT  product_line,
    ROUND((SUM(total - cogs) / SUM(total)) * 100,
        2) AS profit_margin_percent
FROM amazon_sales
GROUP BY product_line
ORDER BY profit_margin_percent DESC;
```

product_line	profit_margin_percent
Food and beverages	4.76
Health and beauty	4.76
Sports and travel	4.76
Fashion accessories	4.76
Home and lifestyle	4.76
Electronic accessories	4.76

Explanation:
This query measures profitability efficiency by calculating profit as a percentage of revenue for each product line.

20: Which branch is MOST EFFICIENT (profit per sale)

```
SELECT
    branch,
    ROUND(SUM(total - cogs) / COUNT(invoice_id), 2) AS profit_per_sale
FROM
    amazon_sales
GROUP BY branch
ORDER BY profit_per_sale DESC;
```

branch	profit_per_sale
C	16.05
B	15.23
A	14.87

Explanation:
This query identifies which branch generates the highest profit per transaction, indicating operational efficiency.

4 Product Performance Analysis

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

```
WITH product_revenue AS (
  SELECT product_line, SUM(total) AS revenue
  FROM amazon_sales
  GROUP BY product_line
)
SELECT product_line, revenue,
    CASE
      WHEN revenue > (SELECT AVG(revenue) FROM product_revenue)
      THEN 'Good'
      ELSE 'Bad'
    END AS performance
FROM product_revenue;
```

product_line	total_sales	performance
Food and beverages	56144.96	Good
Health and beauty	49193.84	Bad
Sports and travel	55123.00	Good
Fashion accessories	54306.03	Good
Home and lifestyle	53861.96	Good
Electronic accessories	54337.64	Good

Explanation:
This query compares each product line's sales against the average sales and classifies performance as "Good" or "Bad".

22: Calculate the average rating for each product line.

```
SELECT
        product_line, AVG(rating) AS avg_rating
FROM
        amazon_sales
GROUP BY product_line
ORDER BY avg_rating DESC;
```

product_line	avg_rating
Food and beverages	7.11322
Fashion accessories	7.02921
Health and beauty	7.00329
Electronic accessories	6.92471
Sports and travel	6.91627
Home and lifestyle	6.83750

Explanation:

This query calculates the average customer rating for each product line, reflecting customer satisfaction levels.

23: Top 3 product lines per branch

```
WITH ranked_products AS (
    SELECT
        branch,  product_line,  SUM(total) AS revenue,
        RANK() OVER (
            PARTITION BY branch
            ORDER BY SUM(total) DESC
        ) AS rnk
    FROM amazon_sales
    GROUP BY branch, product_line )
SELECT *
FROM ranked_products  WHERE rnk <= 3;
```

branch	product_line	revenue	rnk
A	Home and lifestyle	22417.21	1
A	Sports and travel	19372.75	2
A	Electronic accessories	18317.13	3
B	Sports and travel	19988.26	1
B	Health and beauty	19980.70	2
B	Home and lifestyle	17549.21	3
C	Food and beverages	23766.88	1
C	Fashion accessories	21560.11	2
C	Electronic accessories	18968.99	3

Explanation:

This query uses window functions to identify the top three revenue-generating product lines within each branch.

5 Customer Behavior & Segmentation Analysis

24: Identify the branch that exceeded the average number of products sold.

```
SELECT
        branch, SUM(quantity) AS total_quantity
FROM
        amazon_sales
GROUP BY branch
HAVING total_quantity > (SELECT
                AVG(branch_quantity)
        FROM
                (SELECT
                        SUM(quantity) AS branch_quantity
                FROM
                        amazon_sales
                GROUP BY branch) t);
```

branch	total_quantity
A	1859

Explanation:
This query identifies branches whose total quantity sold is higher than the overall average, highlighting strong-performing branches.

25: Which product line is most frequently associated with each gender?

```
WITH gender_product_count AS (
        SELECT
                gender,  product_line,  COUNT(*) AS purchase_count
        FROM amazon_sales
        GROUP BY gender, product_line),
ranked_products AS (
        SELECT
                gender, product_line,  purchase_count,
                RANK() OVER (
                        PARTITION BY gender
                        ORDER BY purchase_count DESC
                ) AS rnk
        FROM gender_product_count)
SELECT
        gender, product_line,  purchase_count
FROM ranked_products
WHERE rnk = 1;
```

gender	product_line	purchase_count
Female	Fashion accessories	96
Male	Health and beauty	88

Explanation:

This query determines which product line is most frequently purchased by each gender, helping understand gender-based preferences.

26: Which customer type makes the most purchases?

```
SELECT
    customer_type, COUNT(invoice_id) AS invoice_count
FROM
    amazon_sales
GROUP BY customer_type
ORDER BY invoice_count DESC;
```

customer_type	invoice_count
Member	501
Normal	499

Explanation:
This query finds which customer type appears most often in transactions, showing dominant customer segments.

27: Determine the predominant gender among customers.

```
SELECT
    gender, COUNT(*) AS customer_count
FROM
    amazon_sales
GROUP BY gender
ORDER BY customer_count DESC
LIMIT 1;
```

gender	customer_count
Female	501

Explanation:
This query identifies the gender with the highest number of purchases, showing dominant customer demographics.

28: Examine the distribution of genders within each branch.

```
SELECT
    branch, gender, COUNT(*) AS customer_count
FROM
    amazon_sales
GROUP BY branch , gender
ORDER BY branch , gender;
```

branch	gender	customer_count
A	Female	161
A	Male	179
B	Female	162
B	Male	170
C	Female	178
C	Male	150

Explanation:
This query shows how male and female customers are distributed across different branches.

29: Does higher rating lead to higher revenue?

```
SELECT
    CASE
        WHEN rating >= 4.5 THEN 'Excellent'
        WHEN rating >= 3.5 THEN 'Good'
        ELSE 'Poor'
    END AS rating_bucket,
    ROUND(SUM(total), 2) AS revenue
FROM
    amazon_sales
GROUP BY rating_bucket
ORDER BY revenue DESC;
```

rating_bucket	revenue
Excellent	293441.09
Good	29526.34

Explanation:
This query groups ratings into categories and compares revenue, helping analyze whether better ratings lead to higher sales.

30: Average basket size (quantity per transaction)

```
SELECT
    ROUND(AVG(quantity), 2) AS avg_items_per_purchase
FROM
    amazon_sales;
```

avg_items_per_purchase
5.51

Explanation:
This query calculates the average number of items purchased per transaction, indicating customer buying behavior.

6 Time-Based Sales & Rating Analysis

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

```
SELECT
    day_name, time_of_day, COUNT(invoice_id) AS sales_count
FROM amazon_sales
GROUP BY day_name , time_of_day
ORDER BY FIELD(day_name,
        'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday',
        'Sunday') , FIELD(time_of_day, 'Morning', 'Afternoon', 'Evening');
```

day_name	time_of_day	sales_count
Monday	Morning	21
Monday	Afternoon	75
Monday	Evening	29
Tuesday	Morning	36
Tuesday	Afternoon	71
Tuesday	Evening	51
Wednesday	Morning	22
Wednesday	Afternoon	81
Wednesday	Evening	40
Thursday	Morning	33
Thursday	Afternoon	76
Thursday	Evening	29
Friday	Morning	29
Friday	Afternoon	74
Friday	Evening	36
Saturday	Morning	28
Saturday	Afternoon	81
Saturday	Evening	55
Sunday	Morning	22
Sunday	Afternoon	70
Sunday	Evening	41

Explanation:

This query analyzes how sales volume varies by day of the week and time of day.

32: Identify the time of day when customers provide the most ratings.

```
SELECT
    time_of_day, COUNT(rating) AS rating_count
FROM
    amazon_sales
GROUP BY time_of_day
ORDER BY rating_count DESC
LIMIT 1;
```

time_of_day	rating_count
Afternoon	528

Explanation:
This query identifies when customers are most active in giving feedback, useful for engagement analysis.

33: Identify the day of the week with the highest average ratings.

```
SELECT
    day_name, ROUND(AVG(rating), 2) AS avg_rating
FROM
    amazon_sales
GROUP BY day_name
ORDER BY avg_rating DESC
LIMIT 1;
```

day_name	avg_rating
Monday	7.15

Explanation:
This query determines which day of the week has the highest average customer satisfaction.

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

```
WITH avg_ratings AS (
        SELECT branch, time_of_day, AVG(rating) AS avg_rating
        FROM amazon_sales
        GROUP BY branch, time_of_day ),
ranked_ratings AS (
            SELECT branch, time_of_day, avg_rating, RANK() OVER (
                PARTITION BY branch ORDER BY avg_rating DESC
                ) AS rnk FROM avg_ratings )
SELECT  branch, time_of_day, ROUND(avg_rating, 2) AS highest_avg_rating
FROM ranked_ratings
WHERE rnk = 1  ORDER BY branch;
```

branch	time_of_day	highest_avg_rating
A	Afternoon	7.06
B	Morning	6.89
C	Afternoon	7.10

Explanation:
This query identifies the time of day with the highest average rating for each branch using window functions.

35: Determine the day of the week with the highest average ratings for each branch

```
WITH avg_ratings AS (
        SELECT  branch,  day_name,  AVG(rating) AS avg_rating
        FROM amazon_sales
```

```
    GROUP BY branch, day_name ),

ranked_days AS (

    SELECT  branch, day_name,  avg_rating, RANK() OVER (
                    PARTITION BY branch

    ORDER BY avg_rating DESC  ) AS rnk

    FROM avg_ratings)

SELECT  branch, day_name,    ROUND(avg_rating, 2) AS
highest_avg_rating

FROM ranked_days WHERE rnk = 1

ORDER BY branch;
```

branch	day_name	highest_avg_rating
A	Friday	7.31
B	Monday	7.34
C	Friday	7.28

Explanation:
This query finds the day with the highest average rating for each branch, highlighting customer experience trends.

7 Payment Method Analysis

36: Which payment method occurs most frequently?

```
SELECT

    payment_method, COUNT(*) AS occurence

FROM

    amazon_sales

GROUP BY payment_method

ORDER BY occurence DESC;
```

payment_method	occurence
Ewallet	345
Cash	344
Credit card	311

Explanation:

This query identifies which payment method customers use most often, reflecting payment preference trends.

37: Which payment method generates the highest revenue?

```
SELECT
        payment_method, ROUND(SUM(total), 2) AS total_revenue
FROM
        amazon_sales
GROUP BY payment_method
ORDER BY total_revenue DESC;
```

payment_method	total_revenue
Cash	112206.76
Ewallet	109993.38
Credit card	100767.29

Explanation:
This query identifies which payment method contributes the highest total revenue, indicating the most valuable payment channel.

-- Key Findings and Insights --

Sales & Revenue Findings

1. -- Highest Sales Product Line: Electronic Accessories (Units Sold:971) --
2. -- Highest Revenue Product Line: Food and Beverages ($ 56144.96)--

3. -- Lowest Sales Product Line: Health and Beauty (Unit Sold: 854) --

4. -- Lowest Revenue Product Line: Health and Beauty ($ 49193.84) --

Product & Profitability Findings

1. -- Month With Highest Revenue: January ($ 116292.11) --
2. -- City & Branch With Highest Revenue: Naypyitaw[C] ($ 110568.86)--
3. -- Month With Lowest Revenue: February ($ 97219.58) --
4. -- City & Branch With Lowest Revenue: Mandalay[B] ($ 106198.00) --

Customer Analysis:

1. -- Most Predominant Gender: Female --
2. -- Most Predominant Customer Type: Member --
3. -- Highest Revenue Gender: Female ($ 167883.26) --
4. -- Highest Revenue Customer Type: Member ($ 164223.81) --
5. -- Most Popular Product Line (Male): Health and Beauty --
6. -- Most Popular Product Line (Female): Fashion Accessories --

7. -- Distribution Of Members Based On Gender: Male(240) Female(261) --

8. -- Sales Male: 2641 units --

9. --Sales Female: 2869 units --

Conclusion

This SQL-based analysis of Amazon sales data provides valuable insights into how different factors influence sales and profitability. The project highlights the importance of analyzing not just revenue, but also costs, customer behavior, and product performance to gain a complete business perspective.

The analysis shows that sales performance varies significantly across product lines, branches, and time periods. Certain product lines contribute higher profits despite lower sales volume, while customer ratings and time of purchase also have a noticeable impact on revenue. Additionally, member customers and specific branches play a key role in overall business performance.

This project demonstrates my ability to translate raw data into actionable insights using SQL, making it suitable for entry-level Data Analyst roles.