**SQL Capstone Project**

**1. Understanding the Data**

The dataset consists of 17 columns with a focus on sales transactions. Key columns include:

* branch, city → Geographic data
* product\_line, unit\_price, quantity, total, VAT, gross\_income → Sales and revenue data
* customer\_type, gender → Customer demographic information
* date, time → Temporal data for sales trends
* rating → Customer satisfaction measure

**2. Goals and Objectives**

The primary objectives are:

* Product Analysis
* Sales Analysis
* Customer Analysis

**3. Steps to Implement the Project**

**Step 1: Data Wrangling**

1. **Create a Database and Table:**

CREATE DATABASE amazon;

CREATE TABLE amazon\_sales (

invoice\_id VARCHAR(30) NOT NULL,

branch VARCHAR(5) NOT NULL,

city VARCHAR(30) NOT NULL,

customer\_type VARCHAR(30) NOT NULL,

gender VARCHAR(10) NOT NULL,

product\_line VARCHAR(100) NOT NULL,

unit\_price DECIMAL(10,2) NOT NULL,

quantity INT NOT NULL,

VAT FLOAT(6,4) NOT NULL,

total DECIMAL(10,2) NOT NULL,

date DATE NOT NULL,

time TIME NOT NULL,

payment\_method VARCHAR(30) NOT NULL,

cogs DECIMAL(10,2) NOT NULL,

gross\_margin\_percentage FLOAT(11,9) NOT NULL,

gross\_income DECIMAL(10,2) NOT NULL,

rating FLOAT(3,1) NOT NULL

);

1. **Insert Data**
2. **Check for NULL Values:**

SELECT column\_name

FROM information\_schema.columns

WHERE table\_name = 'amazon\_sales' AND is\_nullable = 'YES';

**Step 2: Feature Engineering**

1. **Add a new column named month\_name**

ALTER TABLE amazon\_sales

ADD COLUMN month\_name VARCHAR(20);

UPDATE amazon\_sales SET month\_name = MONTHNAME(date);

1. **Add a new column named time\_of\_day (Extract the part of the day: Morning, Afternoon, or Evening.)**

ALTER TABLE amazon\_sales

ADD COLUMN time\_of\_day VARCHAR(20);

UPDATE amazon\_sales

SET time\_of\_day = CASE

WHEN EXTRACT(HOUR FROM time) BETWEEN 6 AND 11 THEN 'Morning'

WHEN EXTRACT(HOUR FROM time) BETWEEN 12 AND 17 THEN 'Afternoon'

WHEN EXTRACT(HOUR FROM time) BETWEEN 18 AND 23 THEN 'Evening'

ELSE 'Night'

END;

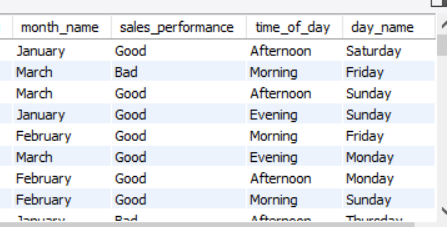
1. **Add a new column named  day\_name (Extract the day of the week.)**

ALTER TABLE amazon\_sales

ADD COLUMN day\_name VARCHAR(10);

UPDATE amazon\_sales

SET day\_name = DAYNAME(date);



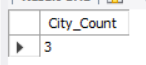
**Step 3: Exploratory Data Analysis (EDA)**

**Business Questions & Queries:**

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

SELECT COUNT(DISTINCT city) AS City\_Count

FROM amazon\_sales;

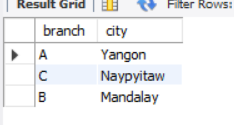


1. **For each branch, what is the corresponding city?**

SELECT branch, city

FROM amazon\_sales

GROUP BY branch, city;



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

SELECT COUNT(DISTINCT product\_line) AS No\_Of\_ProductLine

FROM amazon\_sales;



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

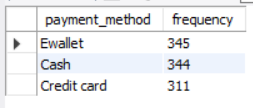
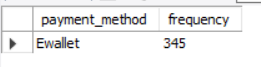
SELECT payment\_method, COUNT(payment\_method) AS frequency

FROM amazon\_sales

GROUP BY payment\_method

ORDER BY frequency DESC

LIMIT 1;



1. **Which product line has the highest sales?**

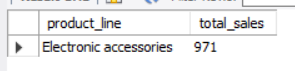
SELECT product\_line, SUM(quantity) AS total\_sales

FROM amazon\_sales

GROUP BY product\_line

ORDER BY total\_sales DESC

LIMIT 1;



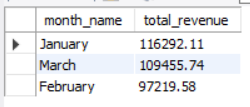
1. **How much revenue is generated each month?**

SELECT month\_name, SUM(total) AS total\_revenue

FROM amazon\_sales

GROUP BY month\_name

ORDER BY total\_revenue DESC



1. **In which month did the cost of goods sell reach its peak?**

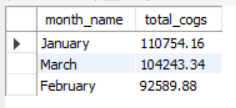
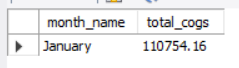
SELECT month\_name, SUM(cogs) AS total\_cogs

FROM amazon\_sales

GROUP BY month\_name

ORDER BY total\_cogs DESC

LIMIT 1;



1. **Which product line generated the highest revenue?**

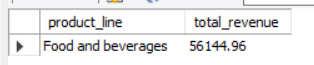
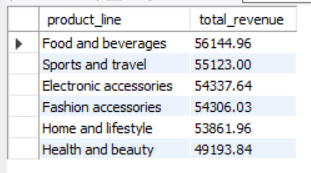
SELECT product\_line, SUM(total) AS total\_revenue

FROM amazon\_sales

GROUP BY product\_line

ORDER BY total\_revenue DESC

LIMIT 1;



1. **In which city was, the highest revenue recorded?**

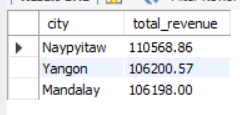
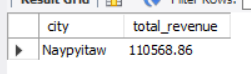
SELECT city, SUM(total) AS total\_revenue

FROM amazon\_sales

GROUP BY city

ORDER BY total\_revenue DESC

LIMIT 1;



1. **Which product line incurred the highest Value Added Tax?**

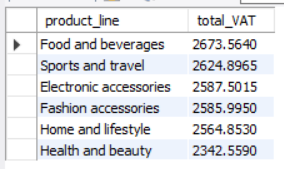
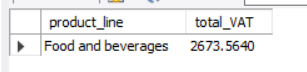
SELECT product\_line, SUM(VAT) AS total\_VAT

FROM amazon\_sales

GROUP BY product\_line

ORDER BY total\_VAT DESC

LIMIT 1;



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

ALTER TABLE amazon\_sales

ADD COLUMN sales\_performance VARCHAR(10);

CREATE TEMPORARY TABLE temp\_avg\_total AS

SELECT AVG(total) AS avg\_total

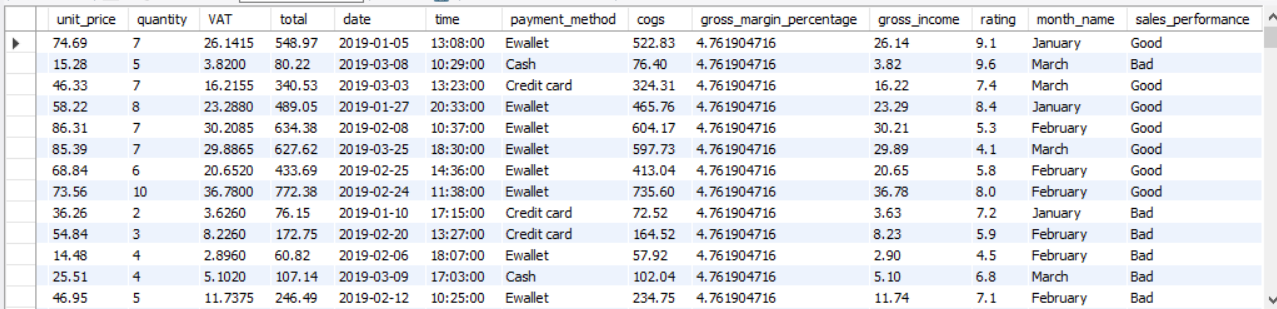
FROM amazon\_sales;

UPDATE amazon\_sales

SET sales\_performance = CASE

WHEN total > (SELECT avg\_total FROM temp\_avg\_total) THEN 'Good'

ELSE 'Bad'

END;

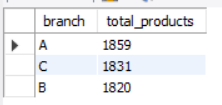
1. **Identify the branch that exceeded the average number of products sold.**

SELECT branch , SUM(quantity) AS total\_products

FROM amazon\_sales

GROUP BY branch

HAVING SUM(quantity) > (SELECT AVG(quantity) FROM amazon\_sales);



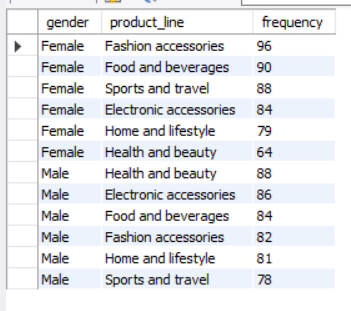
1. **Which product line is most frequently associated with each gender?**

SELECT gender, product\_line, COUNT(\*) AS frequency

FROM amazon\_sales

GROUP BY gender, product\_line

ORDER BY gender, frequency DESC;



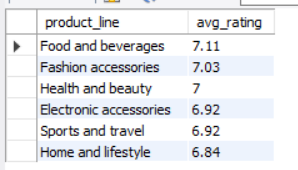
1. **Calculate the average rating for each product line.**

SELECT product\_line, ROUND(AVG(rating),2) AS avg\_rating

FROM amazon\_sales

GROUP BY product\_line

ORDER BY avg\_rating DESC



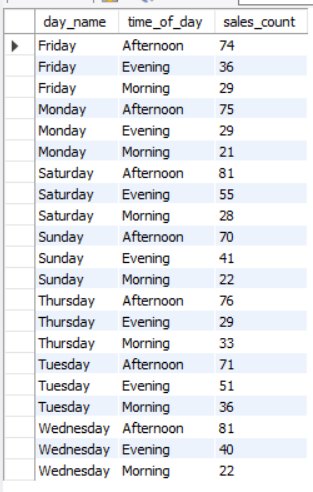
1. **Count the sales occurrences for each time of day on every weekday.**

SELECT day\_name, time\_of\_day, COUNT(\*) AS sales\_count

FROM amazon\_sales

GROUP BY day\_name, time\_of\_day

ORDER BY day\_name, time\_of\_day;



1. **Identify the customer type contributing the highest revenue.**

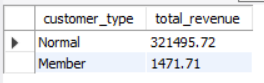
SELECT customer\_type , SUM(total) AS total\_revenue

FROM amazon\_sales

GROUP BY customer\_type

ORDER BY total\_revenue DESC

LIMIT 1;



1. **Determine the city with the highest VAT percentage.**

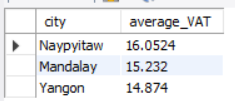
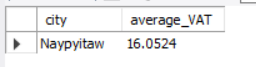
SELECT city, ROUND(AVG(VAT), 4) AS average\_VAT

FROM amazon\_sales

GROUP BY city

ORDER BY average\_VAT DESC

LIMIT 1;



1. **Identify the customer type with the highest VAT payments.**

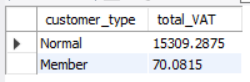
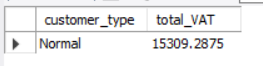
SELECT customer\_type, SUM(VAT) AS total\_VAT

FROM amazon\_sales

GROUP BY customer\_type

ORDER BY total\_VAT DESC

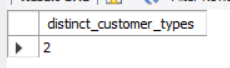
LIMIT 1;



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

SELECT COUNT(DISTINCT customer\_type) AS distinct\_customer\_types

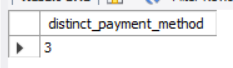
FROM amazon\_sales;



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

SELECT COUNT(DISTINCT payment\_method) AS distinct\_payment\_method

FROM amazon\_sales;



1. **Which customer type occurs most frequently?**

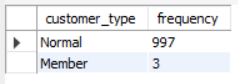
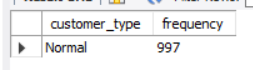
SELECT customer\_type, COUNT(\*) AS frequency

FROM amazon\_sales

GROUP BY customer\_type

ORDER BY frequency DESC

LIMIT 1;



1. **Identify the customer type with the highest purchase frequency.**

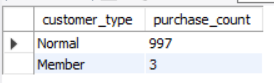
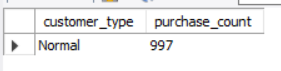
SELECT customer\_type, COUNT(\*) AS purchase\_count

FROM amazon\_sales

GROUP BY customer\_type

ORDER BY purchase\_count DESC

LIMIT 1;



1. **Determine the predominant gender among customers.**

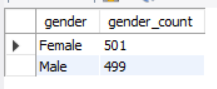
SELECT gender, COUNT(\*) AS gender\_count

FROM amazon\_sales

GROUP BY gender

ORDER BY gender\_count DESC

LIMIT 1;



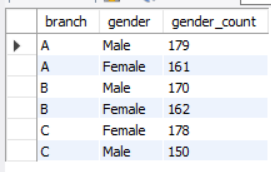
1. **Examine the distribution of genders within each branch.**

SELECT branch, gender, COUNT(\*) AS gender\_count

FROM amazon\_sales

GROUP BY branch, gender

ORDER BY branch, gender\_count DESC;



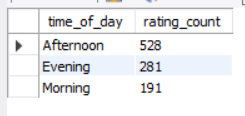
1. **Identify the time of day when customers provide the most ratings.**

SELECT time\_of\_day, COUNT(rating) AS rating\_count

FROM amazon\_sales

GROUP BY time\_of\_day

ORDER BY rating\_count DESC;



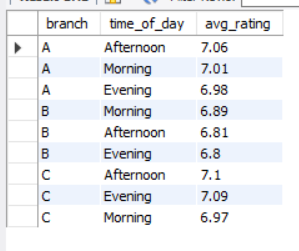
1. **Determine the time of day with the highest customer ratings for each branch.**

SELECT branch, time\_of\_day, ROUND(AVG(rating), 2) AS avg\_rating

FROM amazon\_sales

GROUP BY branch, time\_of\_day

ORDER BY branch, avg\_rating DESC;



1. **Identify the day of the week with the highest average ratings.**

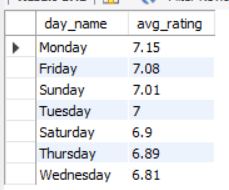
SELECT day\_name, ROUND(AVG(rating), 2) AS avg\_rating

FROM amazon\_sales

GROUP BY day\_name

ORDER BY avg\_rating DESC

LIMIT 1;



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

SELECT branch, day\_name, ROUND(AVG(rating), 2) AS avg\_rating

FROM amazon\_sales

GROUP BY branch, day\_name

ORDER BY branch, avg\_rating DESC;

