****a

**About the Project**

This project uses SQL to create and manage a database of Walmart sales data. It involves table creation, data cleaning, and the transformation of columns—for example, adding "time\_of\_day", "day\_name", and "month\_name"—to enable detailed analysis. The queries investigate sales trends, customer behavior, product performance, and tax-related metrics across different dimensions.

**Key Insights**

* **Unique Cities:** The dataset covers three cities: Yangon, Naypyitaw, and Mandalay.
* **Branch Locations:** Branch A is in Yangon, branch B in Mandalay, and branch C in Naypyitaw.
* **Product Lines:** There are six distinct product lines ranging from food and beverages to electronic accessories.
* **Top Selling Product Line:** Electronic accessories lead in quantity sold.
* **Monthly Revenue:** January generates the highest revenue, followed by March and then February.
* **Largest COGS:** January also records the highest cost of goods sold.
* **Product Revenue Leader:** Food and beverages generate the largest overall revenue.
* **City Revenue:** Naypyitaw leads in total revenue compared to Yangon and Mandalay.
* **Highest VAT:** Food and beverages incur the highest VAT among product lines.
* **Sales Performance Rating:** When comparing to the sales average, all product lines are classified as "Bad" since none exceed the threshold.
* **Branch Sales Volume:** All branches perform above the overall average, with branch A posting the highest total quantity sold.
* **Product Line by Gender:** Females most commonly purchase fashion accessories, whereas males lean slightly toward health and beauty products.
* **Product Ratings:** Average ratings across product lines show little variation, suggesting overall steady customer satisfaction.
* **Customer Types:** The sales include two types of customers—Normal and Member—with Members making just a few more purchases.
* **Payment Methods:** Customers use three payment methods: credit card, e-wallet, and cash.
* **Gender Distribution:** The customer base is nearly even, with a minimal male majority.
* **Ratings by Time of Day:** Customers give their highest ratings in the afternoon, with slightly lower scores in the morning and evening.
* **Daily Ratings:** Monday enjoys the best average ratings, followed by Friday and Tuesday.
* **Sales Volume by Day:** Sales are generally higher in the afternoon and evening, especially on popular days like Monday and Friday.
* **Revenue by Customer Type:** Members contribute a slightly higher total revenue compared to Normal customers.
* **Tax Percent by City:** Naypyitaw has the highest average tax rate, while Yangon records the lowest.
* **VAT by Customer Type:** Members tend to pay a slightly higher average VAT than Normal customers.

-- Create database

CREATE DATABASE IF NOT EXISTS walmartSales;

-- Create table

CREATE TABLE IF NOT EXISTS sales(

    invoice\_id VARCHAR(30) NOT NULL PRIMARY KEY,

    branch VARCHAR(5) NOT NULL,

    city VARCHAR(30) NOT NULL,

    customer\_type VARCHAR(30) NOT NULL,

    gender VARCHAR(30) NOT NULL,

    product\_line VARCHAR(100) NOT NULL,

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

    quantity INT NOT NULL,

    tax\_pct FLOAT(6,4) NOT NULL,

    total DECIMAL(12, 4) NOT NULL,

    date DATETIME NOT NULL,

    time TIME NOT NULL,

    payment VARCHAR(15) NOT NULL,

    cogs DECIMAL(10,2) NOT NULL,

    gross\_margin\_pct FLOAT(11,9),

    gross\_income DECIMAL(12, 4),

    rating FLOAT(2, 1)

);

-- Data cleaning

SELECT

    \*

FROM sales;

-- Add the time\_of\_day column

SELECT

    time,

    (CASE

        WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"

        WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"

        ELSE "Evening"

    END) AS time\_of\_day

FROM sales;

ALTER TABLE sales ADD COLUMN time\_of\_day VARCHAR(20);

-- For this to work turn off safe mode for update

-- Edit > Preferences > SQL Edito > scroll down and toggle safe mode

-- Reconnect to MySQL: Query > Reconnect to server

UPDATE sales

SET time\_of\_day = (

    CASE

        WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"

        WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"

        ELSE "Evening"

    END) ;

-- Add day\_name column

SELECT

    date,

    DAYNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN day\_name VARCHAR(10);

UPDATE sales

SET day\_name = DAYNAME(date);

-- Add month\_name column

SELECT

    date,

    MONTHNAME(date)

FROM sales;

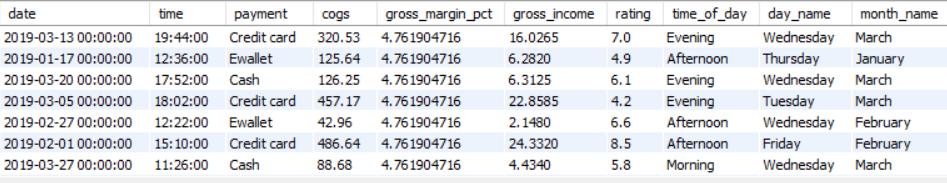
ALTER TABLE sales ADD COLUMN month\_name VARCHAR(10);

UPDATE sales

SET month\_name = MONTHNAME(date);

**SALES TABLE**





**-- --------------------------------------------------------------------**

**-- ---------------------------- Generic Questions ------------------------------**

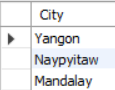
**-- --------------------------------------------------------------------**

-- How many unique cities does the data have?

SELECT

    DISTINCT City

from sales;



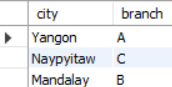
-- In which city is each branch?

SELECT

    DISTINCT city,

    branch

FROM sales;



**-- --------------------------------------------------------------------**

**-- ---------------------------- Product Question -------------------------------**

**-- --------------------------------------------------------------------**

-- How many unique product lines does the data have?

SELECT

    DISTINCT product\_line

FROM sales;



-- Which is the most selling product line

SELECT

    SUM(quantity) as qty,

    product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;



-- What is the total revenue by month

SELECT

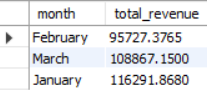
    month\_name AS month,

    SUM(total) AS total\_revenue

FROM sales

GROUP BY month\_name

ORDER BY total\_revenue;



-- which month had the largest COGS?

SELECT

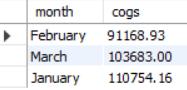
    month\_name AS month,

    SUM(cogs) AS cogs

FROM sales

GROUP BY month\_name

ORDER BY cogs;



-- Which product line had the largest revenue?

SELECT

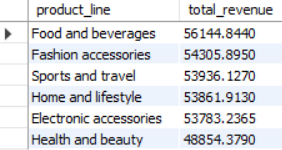
    product\_line,

    SUM(total) as total\_revenue

FROM sales

GROUP BY product\_line

ORDER BY total\_revenue DESC;



-- Which City has the largest revenue?

SELECT

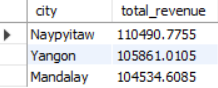
    city,

    SUM(total) AS total\_revenue

FROM sales

GROUP BY city

ORDER BY total\_revenue DESC;



-- Which product line had the largest VAT?

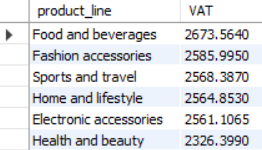
SELECT

    product\_line , SUM(cogs\*0.05) AS VAT

FROM sales

GROUP BY product\_line

ORDER BY vat DESC;



-- Fetch each product line and add a column to those product

-- line showing "Good", "Bad". Good if its greater than average sales

SELECT

    AVG(quantity) AS avg\_qnty

FROM sales;

SELECT

    product\_line,

    CASE

        WHEN AVG(quantity) > 6 THEN "Good"

        ELSE "Bad"

    END AS remark

FROM sales

GROUP BY product\_line;



-- Which branch sold more products than average product sold?

SELECT

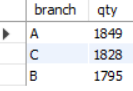
    branch,

    SUM(quantity) AS qty

FROM sales

GROUP BY branch

HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);



-- which is the most common product line by gender ?

SELECT

    gender,

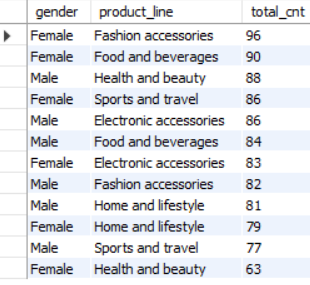
    product\_line,

    COUNT(gender) AS total\_cnt

FROM sales

GROUP BY gender, product\_line

ORDER BY total\_cnt DESC;



-- What is the average rating of each product line

 SELECT

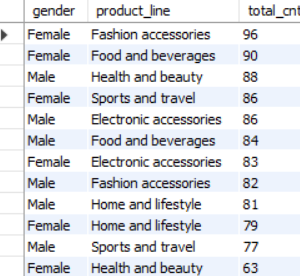
    ROUND(AVG(rating), 2) as avg\_rating,

    product\_line

FROM sales

GROUP BY product\_line

ORDER BY avg\_rating DESC;



**-- --------------------------------------------------------------------**

**-- -------------------------- Customers -------------------------------**

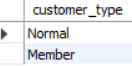
**-- --------------------------------------------------------------------**

-- How many unique customer types does the data have?

SELECT

    DISTINCT customer\_type

FROM sales;

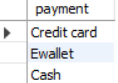


-- How many unique payment methods does the data have?

SELECT

    DISTINCT payment

FROM sales;



-- Which customer type buys the most?

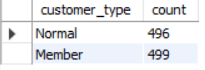
SELECT

    customer\_type,

    COUNT(\*)

FROM sales

GROUP BY customer\_type;



-- What is the gender of most of the customers?

SELECT

    gender,

    COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;



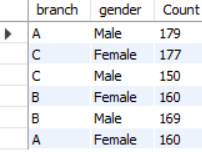
-- What is the gender distribution per branch?

SELECT

    branch,gender, COUNT(gender)

FROM sales

GROUP BY branch, gender;



-- Which time of the day do customers give most ratings?

SELECT

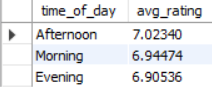
    time\_of\_day,

    AVG(rating) AS avg\_rating

FROM sales

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;



-- Looks like time of the day does not really affect the rating, its

-- more or less the same rating each time of the day.

-- Which time of the day do customers give most ratings per branch?

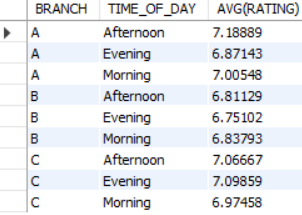
SELECT

    BRANCH , TIME\_OF\_DAY, AVG(RATING)

FROM SALES

GROUP BY BRANCH ,TIME\_OF\_DAY

ORDER BY BRANCH;



-- Which day fo the week has the best avg ratings?

SELECT

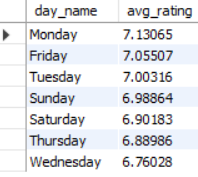
    day\_name,

    AVG(rating) AS avg\_rating

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;



-- Mon, Friday and Tuesday are the top best days for good ratings

-- why is that the case, how many sales are made on these days?

**-- --------------------------------------------------------------------**

**-- ---------------------------- Sales ---------------------------------**

**-- --------------------------------------------------------------------**

-- Number of sales made in each time of the day per weekday

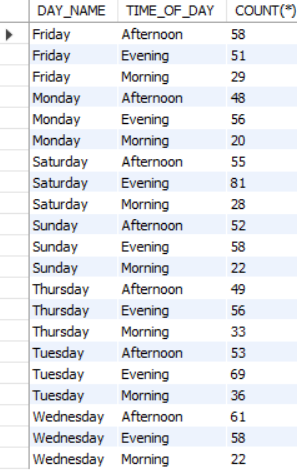
SELECT

    DAY\_NAME,TIME\_OF\_DAY ,COUNT(\*)

FROM SALES

GROUP BY DAY\_NAME,TIME\_OF\_DAY

ORDER BY DAY\_NAME;



-- Which of the customer types brings the most revenue?

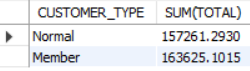
SELECT

    CUSTOMER\_TYPE , SUM(TOTAL)

FROM SALES

GROUP BY CUSTOMER\_TYPE

ORDER BY SUM(TOTAL);



-- Which city has the largest tax/VAT percent?

SELECT

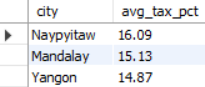
    city,

    ROUND(AVG(tax\_pct), 2) AS avg\_tax\_pct

FROM sales

GROUP BY city

ORDER BY avg\_tax\_pct DESC;



-- Which customer type pays the most in VAT?

SELECT

    customer\_type,

    AVG(tax\_pct) AS total\_tax

FROM sales

GROUP BY customer\_type

ORDER BY total\_tax;

