**Walmart Sales Data Analysis**

**Project Overview**

This project aims to explore the Walmart Sales data to understand top performing branches and products, sales trend of of different products, customer behaviour. The aims is to study how sales strategies can be improved and optimized. The dataset was obtained from the [Kaggle Walmart Sales Forecasting Competition](https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting).

"In this recruiting competition, job-seekers are provided with historical sales data for 45 Walmart stores located in different regions. Each store contains many departments, and participants must project the sales for each department in each store. To add to the challenge, selected holiday markdown events are included in the dataset. These markdowns are known to affect sales, but it is challenging to predict which departments are affected and the extent of the impact."

**Purposes of The Project**

The dataset was obtained from the [Kaggle Walmart Sales Forecasting Competition](https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting). This dataset contains sales transactions from three different branches of Walmart, respectively located in Mandalay, Yangon and Naypyitaw. The data contains 17 columns and 1000 rows:

|  |  |  |
| --- | --- | --- |
| Column | Description | Data Type |
| invoice\_id | Invoice of the sales made | VARCHAR(30) |
| branch | Branch at which sales were made | VARCHAR(5) |
| city | The location of the branch | VARCHAR(30) |
| customer\_type | The type of the customer | VARCHAR(30) |
| gender | Gender of the customer making purchase | VARCHAR(10) |
| product\_line | Product line of the product solf | VARCHAR(100) |
| unit\_price | The price of each product | DECIMAL(10, 2) |
| quantity | The amount of the product sold | INT |
| VAT | The amount of tax on the purchase | FLOAT(6, 4) |
| total | The total cost of the purchase | DECIMAL(10, 2) |
| date | The date on which the purchase was made | DATE |
| time | The time at which the purchase was made | TIMESTAMP |
| payment\_method | The total amount paid | DECIMAL(10, 2) |
| cogs | Cost Of Goods sold | DECIMAL(10, 2) |
| gross\_margin\_percentage | Gross margin percentage | FLOAT(11, 9) |
| gross\_income | Gross Income | DECIMAL(10, 2) |
| rating | Rating | FLOAT(2, 1) |

**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 use measure the effectiveness of each sales strategy the business applies and what modificatoins are needed to gain more sales.

1. Customer Analysis

This analysis aims to uncover the different customers 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.
2. Build a database
3. Create table and insert the data.
4. 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.
5. **Feature Engineering:** This will help use generate some new columns from existing ones.
6. Add a new column named time\_of\_day 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.
7. Add a new column named day\_name 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.
8. Add a new column named month\_name 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.
9. **Exploratory Data Analysis (EDA):** Exploratory data analysis is done to answer the listed questions and aims of this project.
10. **Conclusion:**

**Business Questions To Answer**

**Generic Question**

1. How many unique cities does the data have?
2. In which city is each branch?

**Product**

1. How many unique product lines does the data have?
2. What is the most common payment method?
3. What is the most selling product line?
4. What is the total revenue by month?
5. What month had the largest COGS?
6. What product line had the largest revenue?
7. What is the city with the largest revenue?
8. What product line had the largest VAT?
9. Fetch each product line and add a column to those product line showing "Good", "Bad". Good if its greater than average sales
10. Which branch sold more products than average product sold?
11. What is the most common product line by gender?
12. What is the average rating of each product line?

**Sales**

1. Number of sales made in each time of the day per weekday
2. Which of the customer types brings the most revenue?
3. Which city has the largest tax percent/ VAT (**Value Added Tax**)?
4. Which customer type pays the most in VAT?

**Customer**

1. How many unique customer types does the data have?
2. How many unique payment methods does the data have?
3. What is the most common customer type?
4. Which customer type buys the most?
5. What is the gender of most of the customers?
6. What is the gender distribution per branch?
7. Which time of the day do customers give most ratings?
8. Which time of the day do customers give most ratings per branch?
9. Which day fo the week has the best avg ratings?
10. Which day of the week has the best average ratings per branch?

**Revenue And Profit Calculations**

COGS = unitsPrice \* quantity

VAT = 5% \* COGS

VAT is added to the COGS and this is what is billed to the customer.

total(gross\_sales) = VAT + COGS

grossProfit(grossIncome) = total(gross\_sales) - COGS

**Gross Margin** is gross profit expressed in percentage of the total(gross profit/revenue)

text{Gross Margin} = frac{\text{gross income}}{\text{total revenue}}

**Code**

-- Create database

CREATE DATABASE IF NOT EXISTS walmartSales;

use 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

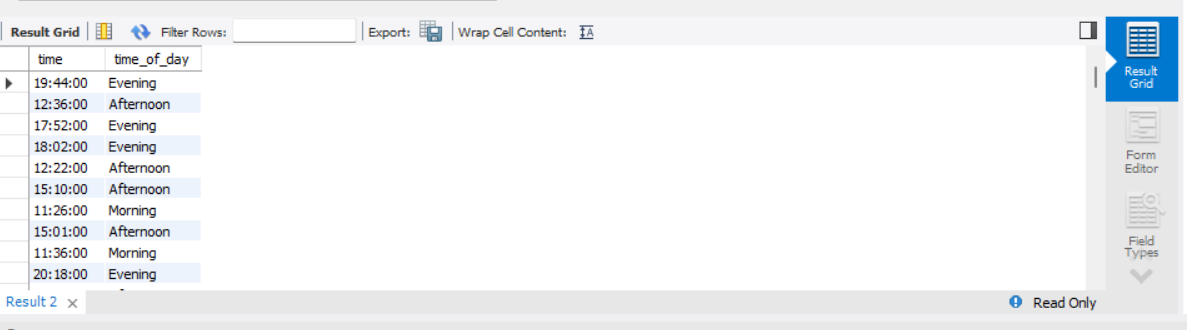
WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"

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

);

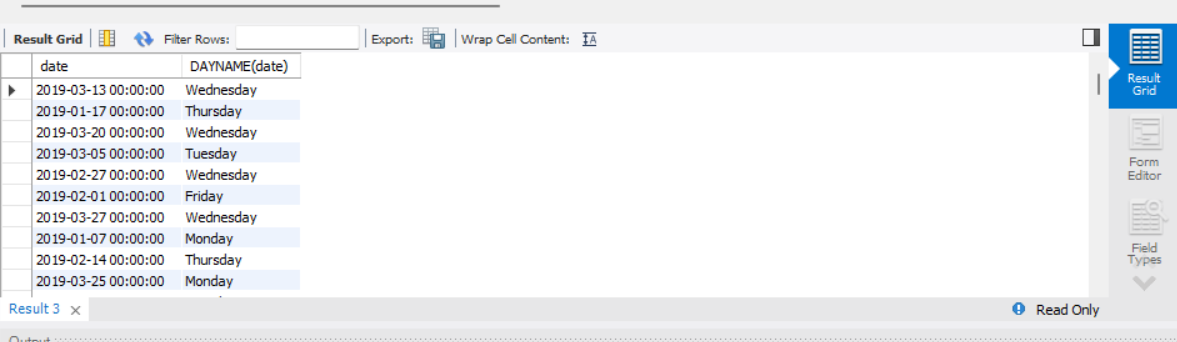
-- Add day\_name column

SELECT

date,

DAYNAME(date)

FROM sales;



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

UPDATE sales

SET day\_name = DAYNAME(date);

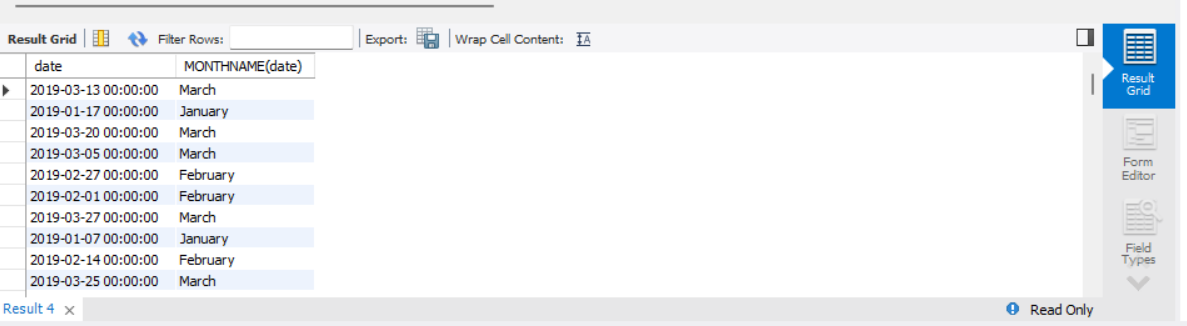
-- **Add month\_name column**

SELECT

date,

MONTHNAME(date)

FROM sales;



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

UPDATE sales

SET month\_name = MONTHNAME(date);

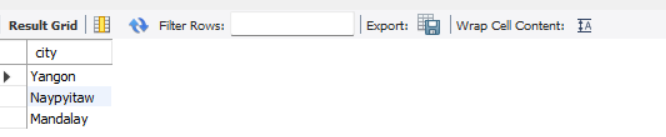
-- ------------------------Generic--------------------------------------------

-- **How many unique cities does the data have?**

SELECT

DISTINCT city

FROM sales;



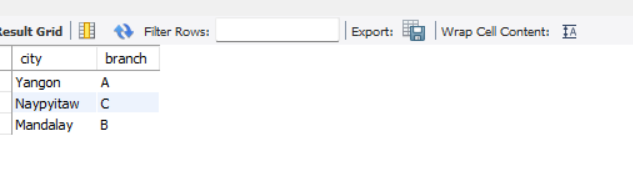
-- **In which city is each branch?**

SELECT

DISTINCT city,

branch

FROM sales;



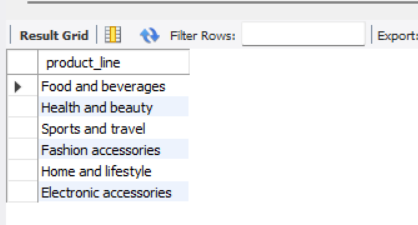
-- ---------------------------- Product -------------------------------

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

SELECT

DISTINCT product\_line

FROM sales;



-- **What is the most selling product line**

SELECT

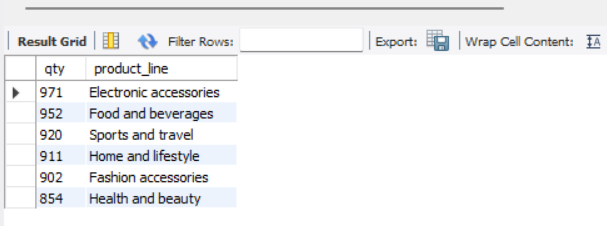
SUM(quantity) as qty,

product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;



-- **What is the most selling product line**

SELECT

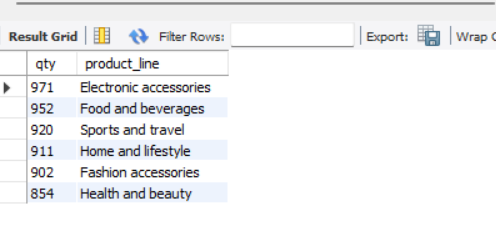
SUM(quantity) as qty,

product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;



-- **What is the total revenue by month**

SELECT

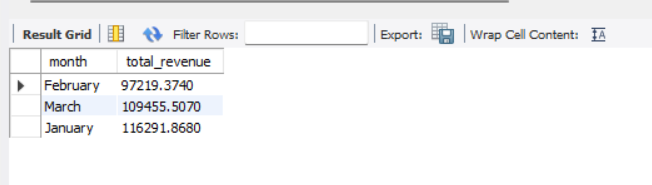
month\_name AS month,

SUM(total) AS total\_revenue

FROM sales

GROUP BY month\_name

ORDER BY total\_revenue;



-- **What month had the largest COGS?**

SELECT

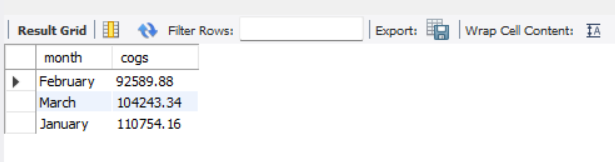
month\_name AS month,

SUM(cogs) AS cogs

FROM sales

GROUP BY month\_name

ORDER BY cogs;



-- **What product line had the largest revenue?**

SELECT

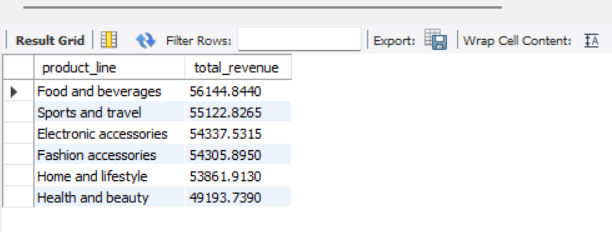
product\_line,

SUM(total) as total\_revenue

FROM sales

GROUP BY product\_line

ORDER BY total\_revenue DESC;



-- **What is the city with the largest revenue?**

SELECT

branch,

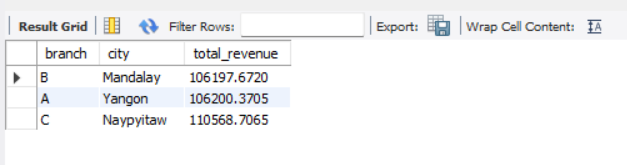
city,

SUM(total) AS total\_revenue

FROM sales

GROUP BY city, branch

ORDER BY total\_revenue;



-- **What product line had the largest VAT?**

SELECT

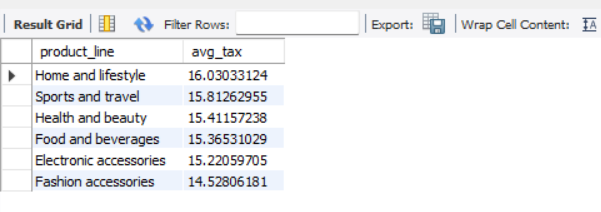
product\_line,

AVG(tax\_pct) as avg\_tax

FROM sales

GROUP BY product\_line

ORDER BY avg\_tax 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

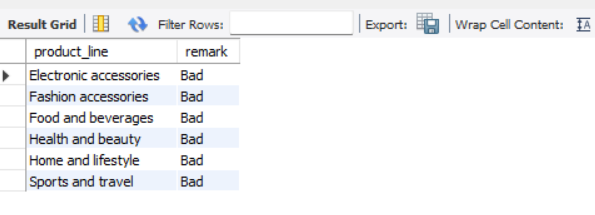
WHEN AVG(quantity) > 6 THEN "Good"

ELSE "Bad"

END AS remark

FROM sales

GROUP BY product\_line;



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

SELECT

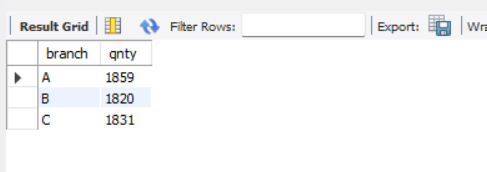
branch,

SUM(quantity) AS qnty

FROM sales

GROUP BY branch

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



-- **What is the most common product line by gender**

SELECT

gender,

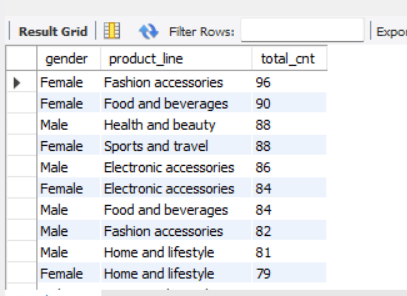
product\_line,

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

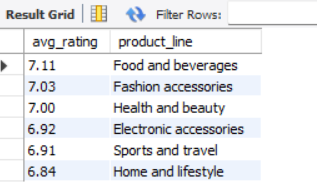
ROUND(AVG(rating), 2) as avg\_rating,

product\_line

FROM sales

GROUP BY product\_line

ORDER BY avg\_rating DESC;



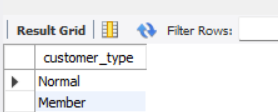
-- -------------------------- Customers -------------------------------

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

SELECT

DISTINCT customer\_type

FROM sales;

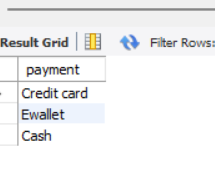


-- **How many unique payment methods does the data have?**

SELECT

DISTINCT payment

FROM sales;



-- **What is the most common customer type?**

SELECT

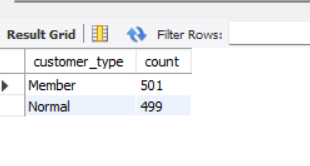
customer\_type,

count(\*) as count

FROM sales

GROUP BY customer\_type

ORDER BY count DESC;



-- **Which customer type buys the most?**

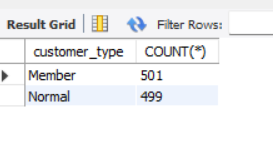
SELECT

customer\_type,

COUNT(\*)

FROM sales

GROUP BY customer\_type;



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

SELECT

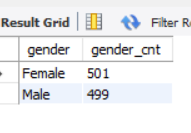
gender,

COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;



-- **What is the gender distribution per branch?**

SELECT

gender,

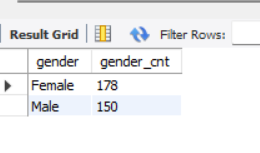
COUNT(\*) as gender\_cnt

FROM sales

WHERE branch = "C"

GROUP BY gender

ORDER BY gender\_cnt DESC;



-- **Gender per branch is more or less the same hence, I don't think has**

**-- an effect of the sales per branch and other factors.**

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

SELECT

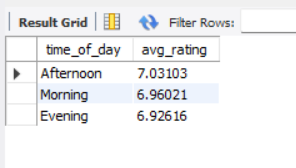
time\_of\_day,

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.alter**

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

SELECT

time\_of\_day,

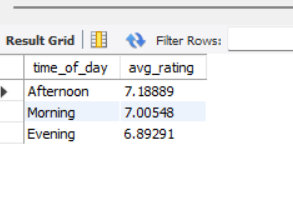
AVG(rating) AS avg\_rating

FROM sales

WHERE branch = "A"

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;



-- **Branch A and C are doing well in ratings, branch B needs to do a**

**-- little more to get better ratings.**

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

SELECT

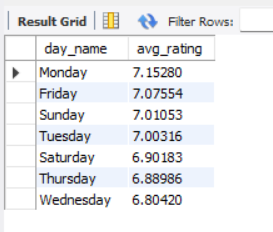
day\_name,

AVG(rating) AS avg\_rating

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;



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

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

**-- Which day of the week has the best average ratings per branch?**

SELECT

day\_name,

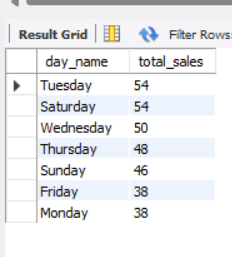
COUNT(day\_name) total\_sales

FROM sales

WHERE branch = "C"

GROUP BY day\_name

ORDER BY total\_sales DESC;



-- ---------------------------- Sales ---------------------------------

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

SELECT

time\_of\_day,

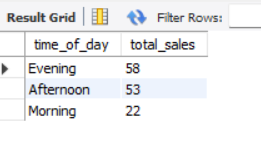
COUNT(\*) AS total\_sales

FROM sales

WHERE day\_name = "Sunday"

GROUP BY time\_of\_day

ORDER BY total\_sales DESC;



-- **Evenings experience most sales, the stores are**

**-- filled during the evening hours**

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

SELECT

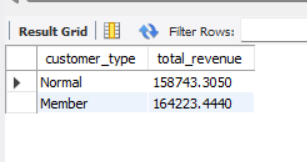
customer\_type,

SUM(total) AS total\_revenue

FROM sales

GROUP BY customer\_type

ORDER BY total\_revenue;



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

SELECT

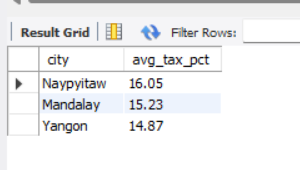
city,

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;

