**Walmart Sales Data Analysis**

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 major aim of thie project is to gain insight into the sales data of Walmart to understand the different factors that affect sales of the different branches.

**About Data**

This dataset contains sales transactions from a 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 sold | 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) |

**RESULTS AND DISCUSSIONS**

* 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.

* 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.
* Customer Analysis
* This analysis aims to uncover the different customers segments, purchase trends and the profitability of each customer segment.

**Approach Used and Query**

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.

Build a database

Create table and insert the data.

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.

CREATE DATABASE IF NOT EXISTS walmartSales1;

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)

);

SELECT

\*

FROM sales;

* Click table and click import and import the data walmart

Feature Engineering: This will help use generate some new columns from existing ones.

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.

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);

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.

SELECT

date,

DAYNAME(date)

FROM sales;

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

UPDATE sales

SET day\_name = DAYNAME(date);

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.

SELECT

date,

MONTHNAME(date)

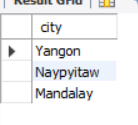
FROM sales;

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

UPDATE sales

SET month\_name = MONTHNAME(date);

Exploratory Data Analysis (EDA)

* unique cities does the data have

SELECT DISTINCT city FROM sales;

* Branch in each city
* SELECT DISTINCT city,branch FROM sales;
* Number of unique product lines present in the data
* select distinct product\_line from sales;
* most selling product line

SELECT SUM(quantity) as qty,product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;

* 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 desc;

* The month with the largest COGS

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

FROM sales

GROUP BY month\_name

ORDER BY cogs;

* product line with the largest revenue

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

FROM sales

GROUP BY product\_line

ORDER BY total\_revenue DESC;

* The city with the largest revenue

SELECT branch,city,SUM(total) AS total\_revenue

FROM sales

GROUP BY city, branch

ORDER BY total\_revenue;

* 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

product\_line,

CASE

WHEN AVG(quantity) > 6 THEN "Good"

ELSE "Bad"

END AS remark

FROM sales

GROUP BY product\_line;

* The 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);

* 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;

* 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;

**Sales**

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

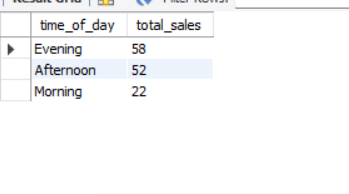
SELECT time\_of\_day,COUNT(\*) AS total\_sales

FROM sales

WHERE day\_name = "Sunday"

GROUP BY time\_of\_day

ORDER BY total\_sales DESC;



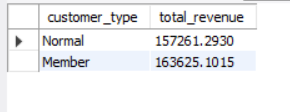
* The customer types brings the most revenue

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

FROM sales

GROUP BY customer\_type

ORDER BY total\_revenue;



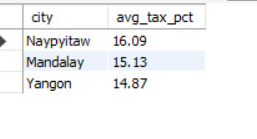
* city with largest tax percent/ VAT (Value Added Tax)

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

FROM sales

GROUP BY city

ORDER BY avg\_tax\_pct DESC;



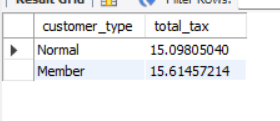
* The customer type pays the most in VAT

SELECT customer\_type,AVG(tax\_pct) AS total\_tax

FROM sales

GROUP BY customer\_type

ORDER BY total\_tax;



* **Customer Analysis**

1. unique customer types in the data

* SELECT DISTINCT customer\_type

FROM sales;

1. unique payment methods in the data

* SELECT DISTINCT payment

FROM sales;

1. Most common type of customer

* SELECT customer\_type,count(\*) as count

FROM sales

GROUP BY customer\_type

ORDER BY count DESC;

1. Type of customer who buys the most

* SELECTcustomer\_type,COUNT(\*)

FROM sales

GROUP BY customer\_type;

1. The gender most of the customers belongs to

* SELECT gender,COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;

1. The gender distribution per branch

* SELECT gender,COUNT(\*) as gender\_cnt

FROM sales

WHERE branch = "C"

GROUP BY gender

ORDER BY gender\_cnt DESC;

1. The time of the day,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;

1. 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;

1. Day fo the week with the best avg ratings

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

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;

1. Day of the week with 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;