

SQL Project

Walmart Sales Analysis

Ankit Jaiswal



Sales Table

```
-- 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)
);
```

Introduction

This project aims to explore the Walmart Sales data to understand top performing branches and products, sales trend of different products, customer behaviour. The aim is to study how sales strategies can be improved and optimized. The dataset was obtained from the [Kaggle Walmart Sales Forecasting Competition](#).





Purposes Of The Project

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

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.



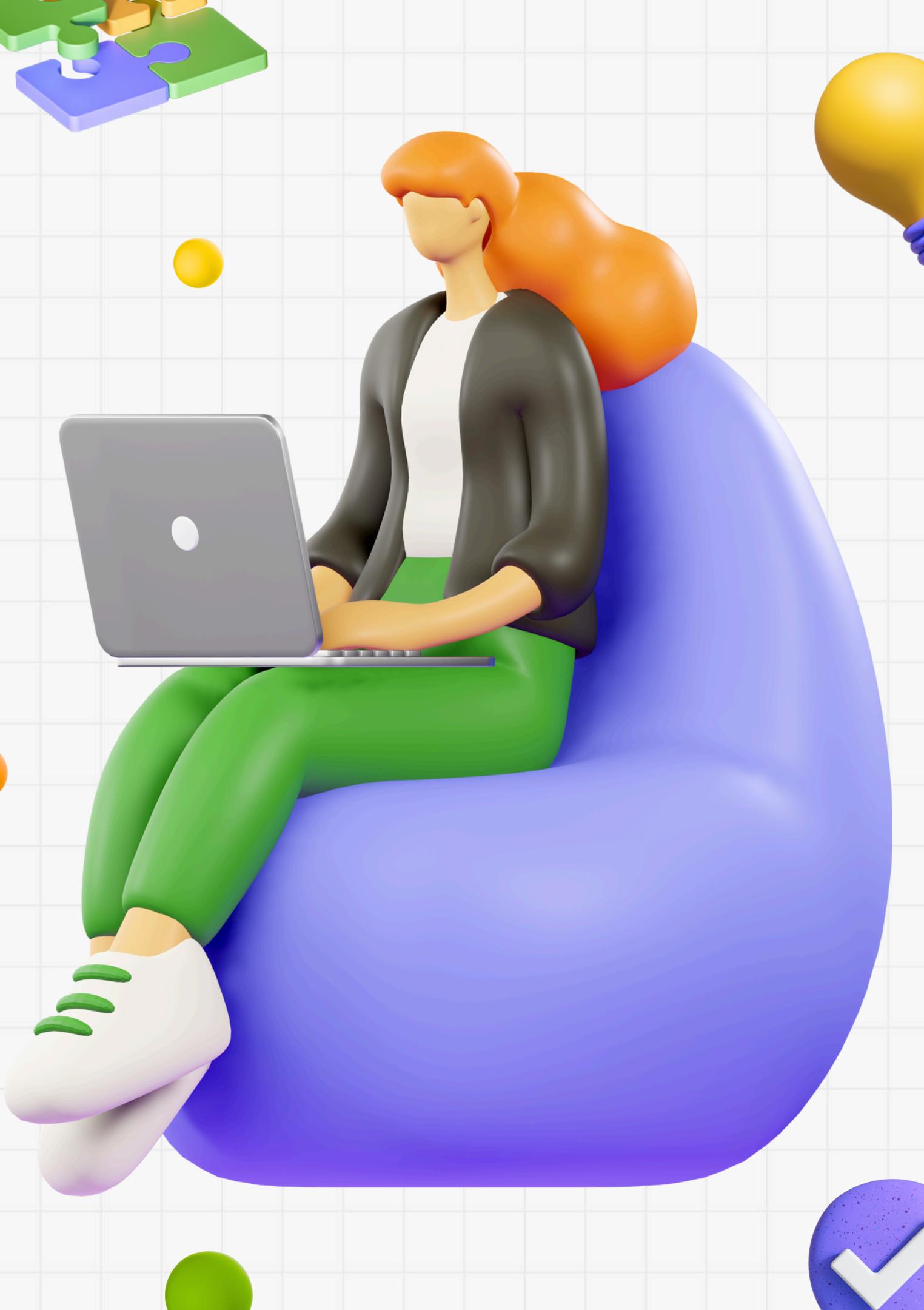
2. Sales Analysis

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 modifications are needed to gain more sales.



3. Customer Analysis

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



Project Approach

1. Data Wrangling

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.

1. Build a database
2. Create table and insert the data.
3. 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.

2. Feature Engineering

1. 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.

```
-- Add the time_of_day column
select time,
       (CASE
            WHEN hour(time) BETWEEN 0 AND 12 THEN "Morning"
            WHEN hour(time) BETWEEN 13 AND 16 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 Editor > 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
);
```



Result Table

time	time_of_day
2023-10-01 08:00:00	Morning
2023-10-01 14:30:00	Evening
2023-10-01 15:00:00	Afternoon
2023-10-01 09:00:00	Evening
2023-10-01 16:00:00	Evening
2023-10-01 13:00:00	Afternoon
2023-10-01 12:00:00	Afternoon
2023-10-01 07:00:00	Morning
2023-10-01 11:00:00	Afternoon

Feature Engineering

2. 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.



**Result column
in Sales Table**

```
SELECT DATE_FORMAT(date, '%W')
FROM sales;

ALTER TABLE sales ADD COLUMN day_name VARCHAR(10);

UPDATE sales
SET day_name = DATE_FORMAT(date, '%W');
```

day_name
Wednesday
Thursday
Wednesday
Tuesday
Wednesday
Friday
Wednesday
Monday

Feature Engineering

3. 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.

```
-- 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);
```



Result Table

month_name
March
January
March
March
February
February
March
January

Business Questions To Answer

Generic Question

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

-- 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;
```

Result Table

	city
▶	Yangon
	Naypyitaw
	Mandalay

	city	branch
▶	Yangon	A
	Naypyitaw	C
	Mandalay	B

Product Analysis

1. How many unique product lines does the data have?

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



Result

COUNT(DISTINCT product_line)
6

2. What is the most common payment method?

```
SELECT  
    payment,  
    COUNT(payment) as cnt  
FROM sales  
GROUP BY payment  
ORDER BY cnt DESC;
```



Result

	payment	cnt
▶	Ewallet	345
	Cash	344
	Credit card	311

Product Analysis

3. What is the most selling product line?

```
SELECT  
    COUNT(quantity) as qty,  
    product_line  
FROM sales  
GROUP BY product_line  
ORDER BY qty DESC;
```



Result

	qty	product_line
▶	178	Fashion accessories
	174	Food and beverages
	170	Electronic accessories
	166	Sports and travel
	160	Home and lifestyle
	152	Health and beauty

4. 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;
```



Result

	month	total_revenue
▶	February	97219.3740
	March	109455.5070
	January	116291.8680

Product Analysis

4. What month had the largest COGS?

```
SELECT  
    month_name AS month,  
    SUM(cogs) AS cogs  
FROM sales  
GROUP BY month_name  
ORDER BY cogs;
```



Result

	month	cogs
▶	February	92589.88
	March	104243.34
	January	110754.16

5. 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;
```



Result

	product_line	total_revenue
▶	Food and beverages	56144.8440
	Sports and travel	55122.8265
	Electronic accessories	54337.5315
	Fashion accessories	54305.8950
	Home and lifestyle	53861.9130
	Health and beauty	49103.7300

Product Analysis

7. 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;
```



Result

	branch	city	total_revenue
▶	B	Mandalay	106197.6720
A	Yangon	106200.3705	
C	Naypyitaw	110568.7065	

8. 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;
```



Result

	product_line	avg_tax
▶	Home and lifestyle	16.03033125
	Sports and travel	15.81262952
	Health and beauty	15.41157237
	Food and beverages	15.36531034
	Electronic accessories	15.22059706

Product Analysis

9. 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;
```



Result

product_line	remark
Food and beverages	Bad
Health and beauty	Bad
Sports and travel	Bad
Fashion accessories	Bad
Home and lifestyle	Bad

10. 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);
```



Result

branch	qnty
A	1859
C	1831
B	1820

Product Analysis

10. 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;
```



Result

	gender	product_line	total_cnt
▶	Female	Fashion accessories	96
	Female	Food and beverages	90
	Male	Health and beauty	88
	Female	Sports and travel	88
	Male	Electronic accessories	86
	Male	Food and beverages	84

12. 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;
```



Result

	avg_rating	product_line
	7.11	Food and beverages
	7.03	Fashion accessories
	7.00	Health and beauty
	6.92	Sports and travel
	6.92	Electronic accessories
	6.84	Home and lifestyle



Sales Analysis

1. 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;
```

Result

time_of_day	total_sales
Evening	58
Afternoon	53
Morning	22



2. 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;
```

Result

customer_type	total_revenue
Normal	158743.3050
Member	164223.4440



Sales Analysis

3. Which city has the 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;
```

Result

city	avg_tax_pct
Naypyitaw	16.05
Mandalay	15.23
Yangon	14.87



4. 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;
```

Result

customer_type	total_tax
Normal	15.14870741
Member	15.60910978





Customer Analysis

1. How many unique customer types does the data have?

```
SELECT  
    COUNT(DISTINCT customer_type) AS cnt  
FROM sales;
```

Result

	cnt
▶	2

2. How many unique payment methods does the data have?

```
SELECT  
    DISTINCT payment  
FROM sales;
```

Result

	payment
▶	Credit card
	Ewallet
	Cash

Customer Analysis

3. What is the most common customer type?

```
SELECT  
    customer_type,  
    count(*) as count  
FROM sales  
GROUP BY customer_type  
ORDER BY count DESC;
```

Result

customer_type	count
Member	501
Normal	499



4. Which customer type buys the most?

```
SELECT  
    customer_type,  
    COUNT(*)  
FROM sales  
GROUP BY customer_type;
```

Result

customer_type	COUNT(*)
Normal	499
Member	501



Customer Analysis

-- 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;
```



Result

	gender	gender_cnt
▶	Female	501
	Male	499



Result

	gender	gender_cnt
▶	Female	178
	Male	150

Customer Analysis

-- 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;**



Result

	day_name	total_sales
1	Tuesday	54
2	Saturday	54
3	Wednesday	50
4	Thursday	48
5	Sunday	46
6	Monday	38

Thank You

For Your Time

Follow Me for More SQL and
Data Science Content

[Follow On Linkedin](#)

