

Walmart Sales Data

Walmart Analysis

MySQL



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 aims are to study how sales strategies can be improved and optimized.

PURPOSE OF THE PROJECT

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

1. ABOUT DATA

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 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)

2. ANALYSIS

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.

3. 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.

- 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.

2. 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.
- 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.
- 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.

3. Exploratory Data Analysis (EDA):

Exploratory data analysis is done to answer the listed questions and aims of this project.

Generic Questions

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

Product Analysis

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?

Customer Analysis

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

Sales Analysis

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?

BUILD DATABASE

CREATE DATABASE IF NOT EXISTS WalmartSales;

USE walmartsales;

Load data from a CSV file into the Sales table

LOAD DATA INFILE 'E:\\WalmartSalesData.csv'

INTO TABLE Sales FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\\n'

IGNORE 1 ROWS;

Select all records from the Sales table to verify the data load

select * from Sales;

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.

-- 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);

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.

```
SELECT
```

```
date,
```

```
DAYNAME(date)
```

```
from sales;
```

```
SELECT * FROM Sales;
```

```
ALTER TABLE Sales ADD COLUMN day_name varchar(12);
```

```
UPDATE Sales
```

```
SET day_name = DAYNAME(date);
```

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.

```
SELECT
```

```
date,
```

```
MONTHNAME(date)
```

```
from sales;
```

```
ALTER TABLE Sales ADD COLUMN month_name varchar(12);
```

```
UPDATE Sales
```

```
SET month_name = MONTHNAME(date);
```

```
SELECT * FROM Sales;
```

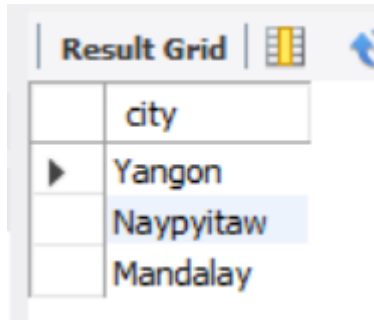
Generic Questions

1. How many unique cities does the data have?

SELECT

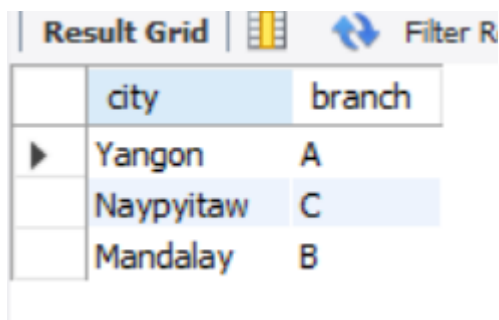
DISTINCT city

from sales;



	city
▶	Yangon
	Naypyitaw
	Mandalay

2. In which city is each branch?



	city	branch
▶	Yangon	A
	Naypyitaw	C
	Mandalay	B

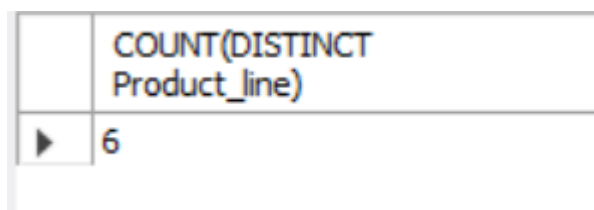
Product Analysis

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

SELECT

COUNT(DISTINCT Product_line)

FROM sales;



	COUNT(DISTINCT Product_line)
▶	6

2. What is the most selling product line?


```

SELECT
SUM(quantity) AS qty,
product_line
FROM Sales
GROUP BY product_line
ORDER BY qty DESC;

```



	qty	product_line
▶	971	Electronic accessories
	952	Food and beverages
	920	Sports and travel
	911	Home and lifestyle
	902	Fashion accessories
	854	Health and beauty

3. What is the total revenue by month?

```

select sum(total) as totalRevenue,
monthname(date) as month
from sales
group by month
Order BY totalRevenue DESC;

```

Result Grid   Filter Rows: <input type="text"/>		
	totalRevenue	month
▶	116291.868000000005	January
	109455.507000000004	March
	97219.373999999997	February

4. What month had the largest COGS?

```

select Sum(cogs) as largestCogs,
month_name

```

from sales

group by month_name

Order BY largestCogs DESC;

	largestCogs	month_name
▶	110754.16000000002	January
	104243.33999999997	March
	92589.88	February

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;

	product_line	total_revenue
▶	Food and beverages	56144.844000000005
	Sports and travel	55122.826499999996
	Electronic accessories	54337.531500000005
	Fashion accessories	54305.895
	Home and lifestyle	53861.91300000001
	Health and beauty	49193.739000000016

6. 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;

	branch	city	total_revenue
▶	B	Mandalay	106197.67199999996
	A	Yangon	106200.3705000001
	C	Naypyitaw	110568.70649999994

7. What product line had the largest VAT?

SELECT product_line,

AVG(Tax) as avg_tax

FROM sales

GROUP BY product_line

ORDER BY avg_tax DESC;

	product_line	avg_tax
▶	Home and lifestyle	16.03033125000001
	Sports and travel	15.812629518072285
	Health and beauty	15.411572368421048
	Food and beverages	15.365310344827583
	Electronic accessories	15.22059705882354
	Fashion accessories	14.528061797752809

8. 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) > (SELECT AVG(quantity) FROM Sales) then
"Good"

else "Bad"

end as remark

from Sales

group by product_line;

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

9. 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);

	branch	Qty
▶	A	1859
	C	1831
	B	1820

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;

	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
	Female	Electronic accessories	84
	Male	Food and beverages	84
	Male	Fashion accessories	82
	Male	Home and lifestyle	81

11. What is the average rating of each product line?

SELECT

product_line,

ROUND(AVG(rating), 2) as avg_rating

FROM sales

GROUP BY product_line

ORDER BY avg_rating DESC;

	product_line	avg_rating
▶	Food and beverages	7.11
	Fashion accessories	7.03
	Health and beauty	7
	Electronic accessories	6.92
	Sports and travel	6.92
	Home and lifestyle	6.84

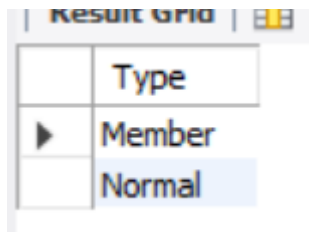
Customer Analysis

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

```
SELECT
```

```
DISTINCT customer_type Type
```

```
FROM sales;
```



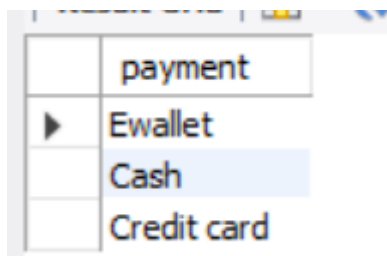
Type
Member
Normal

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

```
SELECT
```

```
DISTINCT payment
```

```
FROM Sales;
```



payment
Ewallet
Cash
Credit card

3. Which customer type buys the most?

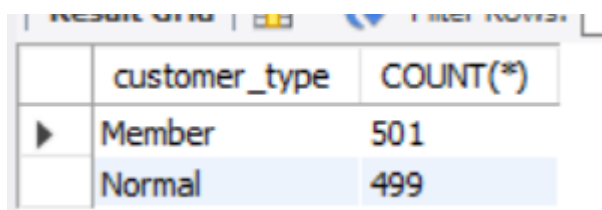
```
SELECT
```

```
customer_type,
```

```
COUNT(*)
```

```
FROM sales
```

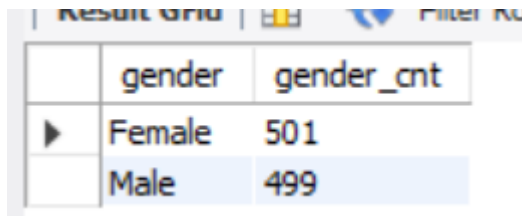
```
GROUP BY customer_type;
```



customer_type	COUNT(*)
Member	501
Normal	499

4. 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;
```

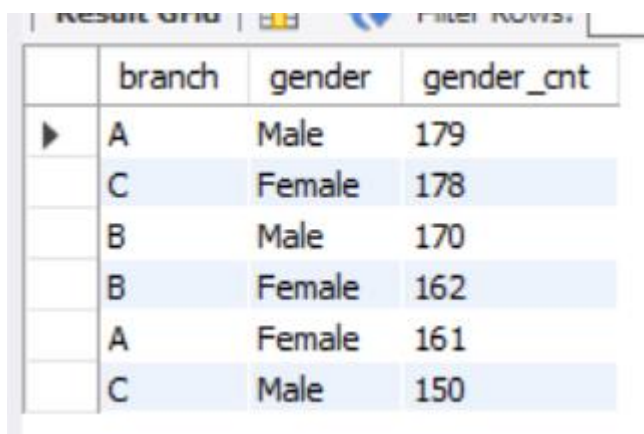


A screenshot of a database query result window. The window has a title bar with 'Result Grid' and a 'Filter Rows' button. The result is displayed in a table with two columns: 'gender' and 'gender_cnt'. The first row shows 'Female' with a count of 501. The second row shows 'Male' with a count of 499.

	gender	gender_cnt
▶	Female	501
	Male	499

5. What is the gender distribution per branch?

```
SELECT  
branch, gender,  
COUNT(gender) as gender_cnt  
FROM sales  
GROUP BY branch, gender  
ORDER BY gender_cnt DESC;
```



A screenshot of a database query result window. The window has a title bar with 'Result Grid' and a 'Filter Rows' button. The result is displayed in a table with four columns: 'branch', 'gender', and 'gender_cnt'. The rows are ordered by 'gender_cnt' in descending order. The data shows that branch A has the highest counts for both genders, followed by branch C, and then branch B.

	branch	gender	gender_cnt
▶	A	Male	179
	C	Female	178
	B	Male	170
	B	Female	162
	A	Female	161
	C	Male	150

6. 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;

```

	time_of_day	avg_rating
▶	Afternoon	7.031299734748012
	Morning	6.960732984293193
	Evening	6.926851851851853

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

```

SELECT
    time_of_day,
    branch,
    AVG(rating) AS avg_rating
FROM sales
WHERE branch IN (select branch from sales)
GROUP BY time_of_day, branch
ORDER BY avg_rating DESC;

```

	time_of_day	branch	avg_rating
▶	Afternoon	A	7.188888888888891
	Evening	C	7.118881118881118
	Afternoon	C	7.066666666666664
	Morning	A	7.005479452054794
	Morning	C	6.974576271186442
	Evening	A	6.893617021276596
	Morning	B	6.891525423728813
	Afternoon	B	6.836799999999998
	Evening	B	6.7729729729729735

8. Which day for 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;
```

	day_name	avg_rating
►	Monday	7.153599999999999
	Friday	7.076258992805756
	Sunday	7.011278195488723
	Tuesday	7.003164556962025
	Saturday	6.901829268292688
	Thursday	6.88985507246377
	Wednesday	6.805594405594405

9. Which day of the week has the best average ratings per branch?

```
SELECT
```

```
    day_name,
```

```
    branch,
```

```
    Avg(rating) as ARB
```

```
FROM sales
```

```
WHERE branch in (select branch from sales)
```

```
GROUP BY day_name, branch
```

```
ORDER BY ARB DESC;
```

	day_name	branch	ARB
►	Monday	B	7.335897435897434
	Friday	A	7.3119999999999985
	Friday	C	7.278947368421051
	Saturday	C	7.229629629629631
	Monday	A	7.097916666666666
	Sunday	A	7.078846153846157
	Wednesday	C	7.064000000000004
	Tuesday	A	7.0588235294117645
	Monday	C	7.036842105263159

Sales Analysis

1. Number of sales made in each time of the day per weekday

SELECT

time_of_day, day_name,

count(invoice) AS total_sales

FROM sales

WHERE day_name Not In ("Saturday", "Sunday")

GROUP BY time_of_day, day_name

ORDER BY total_sales DESC;

	time_of_day	day_name	total_sales
►	Evening	Tuesday	69
	Afternoon	Wednesday	61
	Evening	Wednesday	60
	Afternoon	Friday	58
	Evening	Monday	56
	Evening	Thursday	56
	Afternoon	Tuesday	53
	Evening	Friday	52
	Afternoon	Thursday	49

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;
```

	customer_type	total_revenue
▶	Normal	158743.30500000005
	Member	164223.44400000002

3. Which city has the largest tax percent/ VAT (**Value Added Tax**)?

```
SELECT
city,
ROUND(AVG(tax), 2) AS avg_tax_pct
FROM sales
GROUP BY city
ORDER BY avg_tax_pct DESC;
```

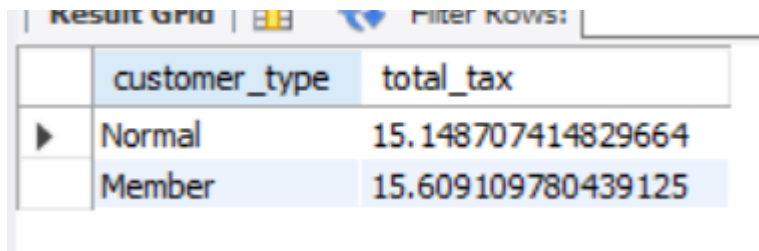
	city	avg_tax_pct
▶	Naypyitaw	16.05
	Mandalay	15.23
	Yangon	14.87

4. Which customer type pays the most in VAT?

```
SELECT
customer_type,
AVG(tax) AS total_tax
FROM sales
```

GROUP BY customer_type

ORDER BY total_tax;



The image shows a screenshot of a SQL query result grid. At the top, there is a header bar with the text "Result Grid" and a "Filter Rows:" dropdown. Below this, the results are displayed in a table with two columns: "customer_type" and "total_tax". The table has two rows: "Normal" with a total tax of 15.148707414829664, and "Member" with a total tax of 15.609109780439125. The "Member" row is highlighted in blue.

	customer_type	total_tax
▶	Normal	15.148707414829664
	Member	15.609109780439125