--first we have to import data file using Right-click on the table you want to insert data into from the left sidebar and select "Table Data Import Wizard"

--Retrive the data using select Statement.

select \* from WalmartSalesdata;

--Generic Question

--Q1. How many unique cities does the data have?

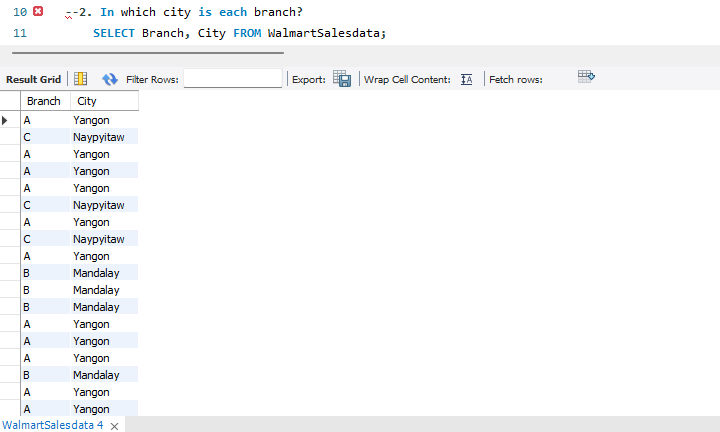
SELECT COUNT(DISTINCT City) AS unique\_city\_count FROM WalmartSalesdata;

A screenshot of a computer

Description automatically generated

--2. In which city is each branch?

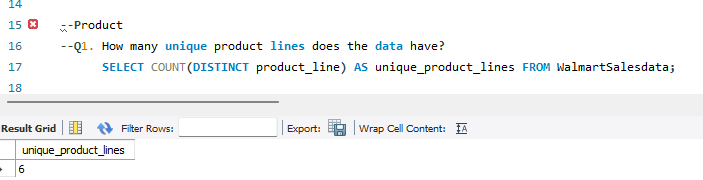
SELECT Branch, City FROM WalmartSalesdata;



--Product

--Q1. How many unique product lines does the data have?

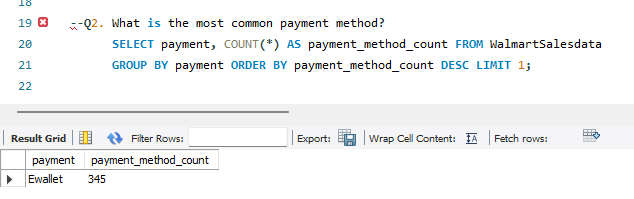
SELECT COUNT(DISTINCT product\_line) AS unique\_product\_lines FROM WalmartSalesdata;



--Q2. What is the most common payment method?

SELECT payment, COUNT(\*) AS payment\_method\_count FROM WalmartSalesdata

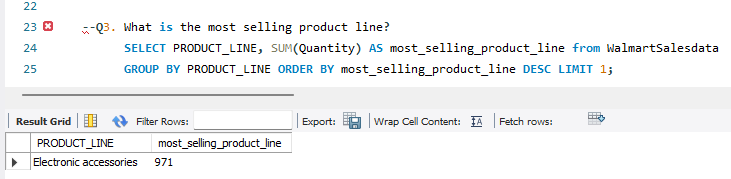
GROUP BY payment ORDER BY payment\_method\_count DESC LIMIT 1;



--Q3. What is the most selling product line?

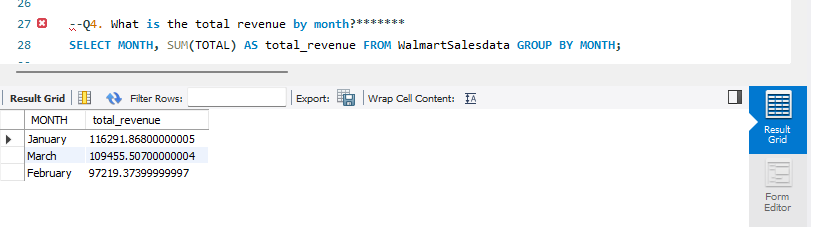
SELECT PRODUCT\_LINE, SUM(Quantity) AS most\_selling\_product\_line from WalmartSalesdata

GROUP BY PRODUCT\_LINE ORDER BY most\_selling\_product\_line DESC LIMIT 1;



--Q4. What is the total revenue by month?\*\*\*\*\*\*\*

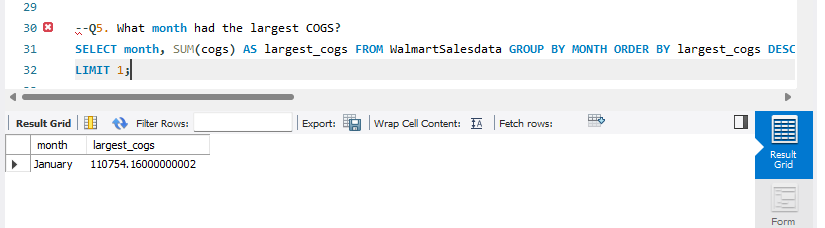
SELECT MONTH, SUM(TOTAL) AS total\_revenue FROM WalmartSalesdata GROUP BY MONTH;



--Q5. What month had the largest COGS?

SELECT month, SUM(cogs) AS largest\_cogs FROM WalmartSalesdata GROUP BY MONTH ORDER BY largest\_cogs DESC

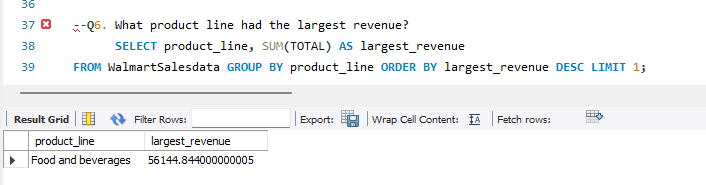
LIMIT 1;



--Q6. What product line had the largest revenue?

SELECT product\_line, SUM(TOTAL) AS largest\_revenue

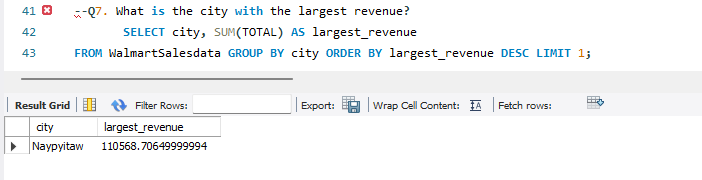
FROM WalmartSalesdata GROUP BY product\_line ORDER BY largest\_revenue DESC LIMIT 1;



--Q7. What is the city with the largest revenue?

SELECT city, SUM(TOTAL) AS largest\_revenue

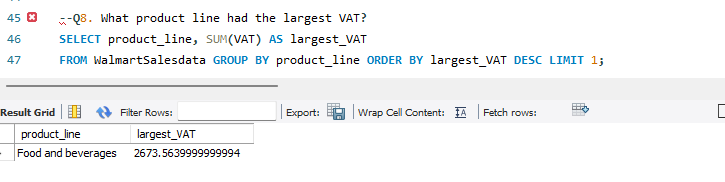
FROM WalmartSalesdata GROUP BY city ORDER BY largest\_revenue DESC LIMIT 1;



--Q8. What product line had the largest VAT?

SELECT product\_line, SUM(VAT) AS largest\_VAT

FROM WalmartSalesdata GROUP BY product\_line ORDER BY largest\_VAT DESC LIMIT 1;



--Q9. 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,

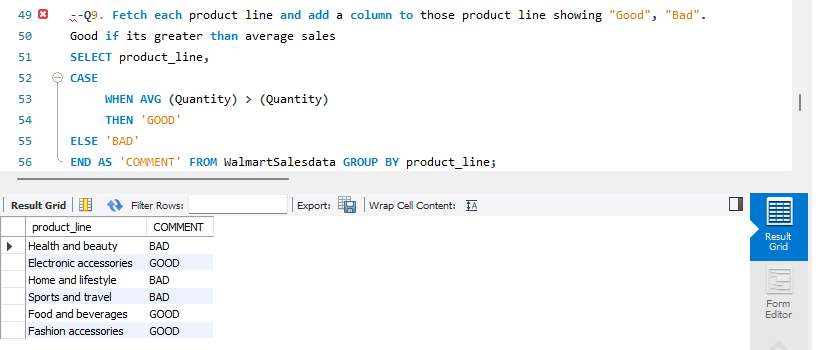
CASE

WHEN AVG (Quantity) > (Quantity)

THEN 'GOOD'

ELSE 'BAD'

END AS 'COMMENT' FROM WalmartSalesdata GROUP BY product\_line;



--Q10. Which branch sold more products than average product sold?

SELECT branch, total\_products\_sold FROM

(SELECT branch, SUM(quantity) AS total\_products\_sold

FROM WalmartSalesdata GROUP BY branch) AS branch\_sales

WHERE

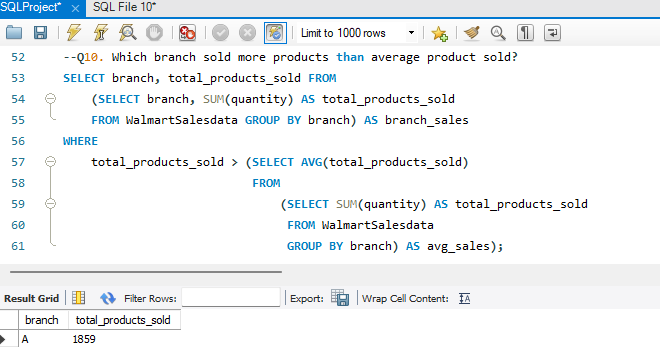
total\_products\_sold > (SELECT AVG(total\_products\_sold)

FROM

(SELECT SUM(quantity) AS total\_products\_sold

FROM WalmartSalesdata

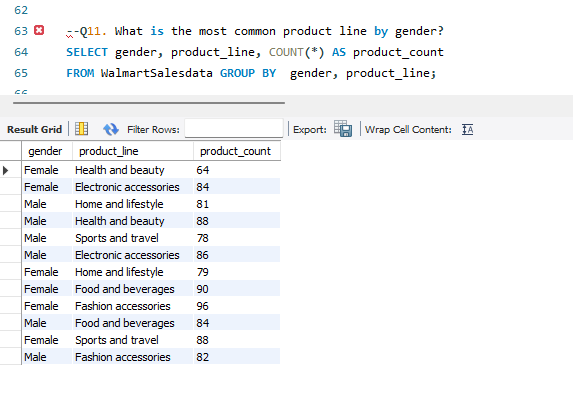
GROUP BY branch) AS avg\_sales);



--Q11. What is the most common product line by gender?

SELECT gender, product\_line, COUNT(\*) AS product\_count

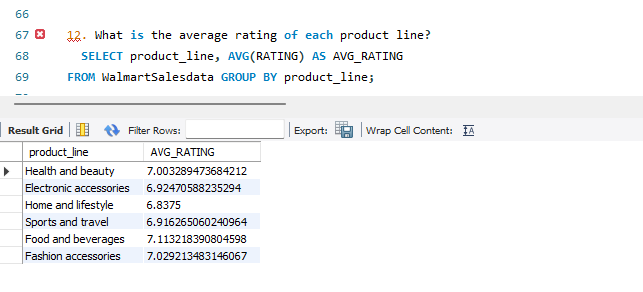
FROM WalmartSalesdata GROUP BY gender, product\_line;



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

SELECT product\_line, AVG(RATING) AS AVG\_RATING

FROM WalmartSalesdata GROUP BY product\_line;



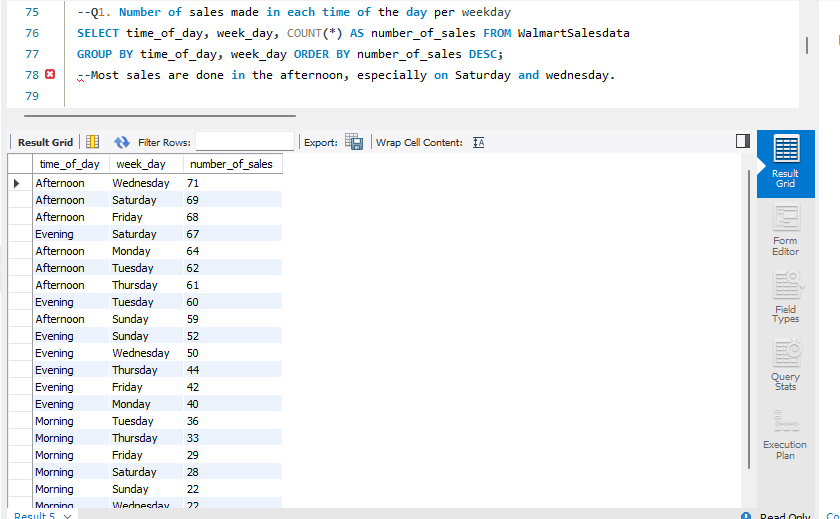
--Sales

--Q1. Number of sales made in each time of the day per weekday

SELECT time\_of\_day, week\_day, COUNT(\*) AS number\_of\_sales FROM WalmartSalesdata

GROUP BY time\_of\_day, week\_day ORDER BY number\_of\_sales DESC;

**--Most sales are done in the afternoon, especially on Saturday and wednesday.**

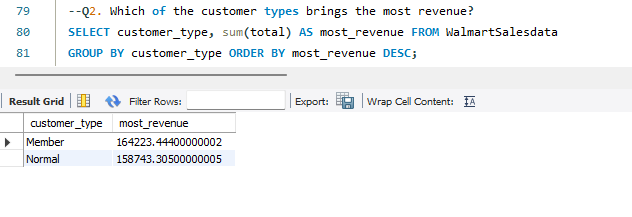


--Q2. Which of the customer types brings the most revenue?

SELECT customer\_type, sum(total) AS most\_revenue FROM WalmartSalesdata

GROUP BY customer\_type ORDER BY most\_revenue DESC;

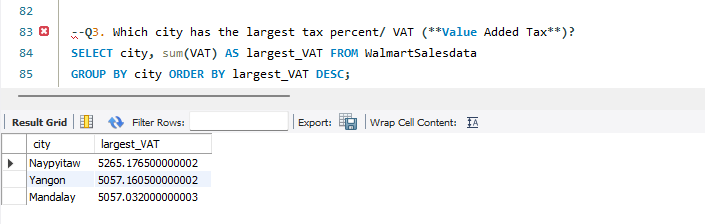
**The bulk of revenue comes from member customer types. However, the margin between it and normal customers is not wide**

--Q3. Which city has the largest tax percent/ VAT (\*\*Value Added Tax\*\*)?

SELECT city, sum(VAT) AS largest\_VAT FROM WalmartSalesdata

GROUP BY city ORDER BY largest\_VAT DESC;

**Naypyitaw has the highest tax rate followed by Mandalay.**

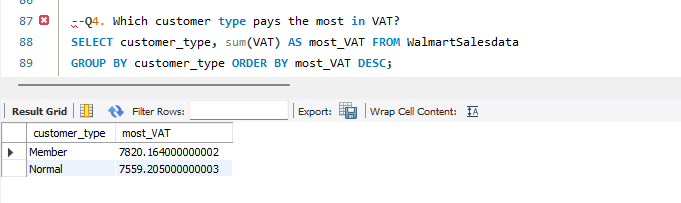


--Q4. Which customer type pays the most in VAT?

SELECT customer\_type, sum(VAT) AS most\_VAT FROM WalmartSalesdata

GROUP BY customer\_type ORDER BY most\_VAT DESC;

**Customers who are members pay the most VAT.**

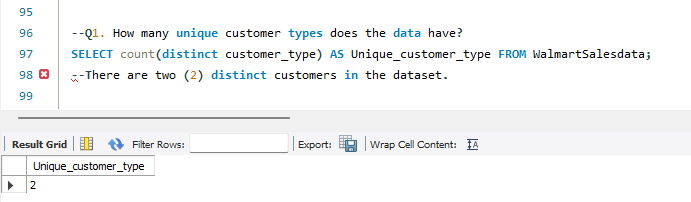


Customer

--Q1. How many unique customer types does the data have?

SELECT count(distinct customer\_type) AS Unique\_customer\_type FROM WalmartSalesdata;

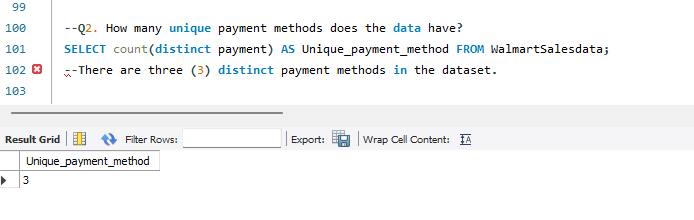
--There are two (2) distinct customers in the dataset.



--Q2. How many unique payment methods does the data have?

SELECT count(distinct payment) AS Unique\_payment\_method FROM WalmartSalesdata;

--There are three (3) distinct payment methods in the dataset.

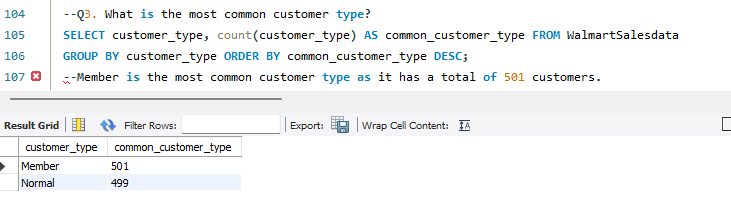


--Q3. What is the most common customer type?

SELECT customer\_type, count(customer\_type) AS common\_customer\_type FROM WalmartSalesdata

GROUP BY customer\_type ORDER BY common\_customer\_type DESC;

--Member is the most common customer type as it has a total of 501 customers.

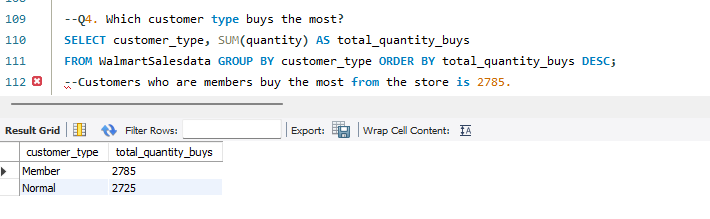


--Q4. Which customer type buys the most?

SELECT customer\_type, SUM(quantity) AS total\_quantity\_buys

FROM WalmartSalesdata GROUP BY customer\_type ORDER BY total\_quantity\_buys DESC;

--Customers who are members buy the most from the store is 2785.

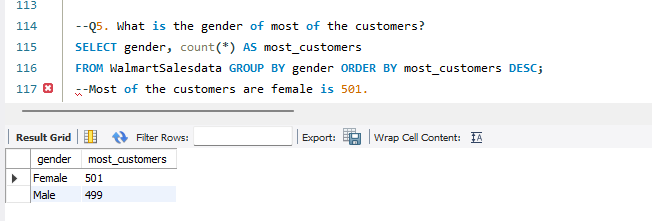


--Q5. What is the gender of most of the customers?

SELECT gender, count(\*) AS most\_customers

FROM WalmartSalesdata GROUP BY gender ORDER BY most\_customers DESC;

--Most of the customers are female is 501.

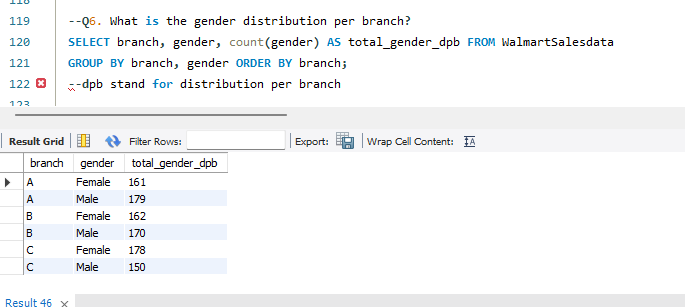


--Q6. What is the gender distribution per branch?

SELECT branch, gender, count(gender) AS total\_gender\_dpb FROM WalmartSalesdata

GROUP BY branch, gender ORDER BY branch;

--dpb stand for distribution per branch



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--WE HAVE SOME TASK PERFORM ON Time\_of\_day, Week\_Day, Month SO WE HAVE TO POPULATING OUR

TABLE WITH SOME NEW COLUMNS

SELECT \* FROM WalmartSalesdata;

--here, I AM CREATING and ADDING the following columns to our database:

Time\_of\_day

Week\_Day

Month

— — CREATING THE COLUMN CALLED time\_of\_day, week\_day, and month USING THE alter statment.

ALTER TABLE WalmartSalesdata

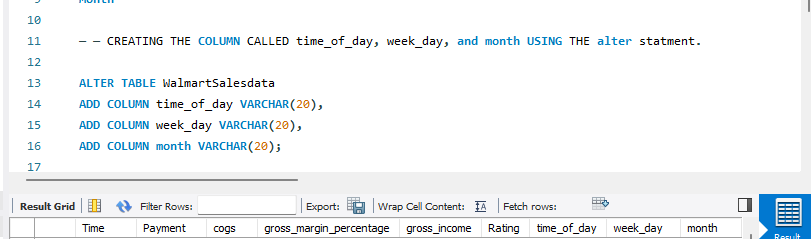
ADD COLUMN time\_of\_day VARCHAR(20),

ADD COLUMN week\_day VARCHAR(20),

ADD COLUMN month VARCHAR(20);

--retriving the data using select statment.

select \* from WalmartSalesdata;



--To create a new column called time\_of\_day based on the time of a transaction using the CASE statement in MySQL, you can use conditions to categorize the time into different parts of the day (e.g., Morning, Afternoon, Evening, Night).

Here's how you can do it:

UPDATE WalmartSalesdata

SET time\_of\_day = CASE

WHEN HOUR(Time) >= 5 AND HOUR(Time) < 12 THEN 'Morning'

WHEN HOUR(Time) >= 12 AND HOUR(Time) < 17 THEN 'Afternoon'

WHEN HOUR(Time) >= 17 AND HOUR(Time) < 21 THEN 'Evening'

ELSE 'Night'

END;

--Explanation

HOUR(transaction\_time): This extracts the hour from the transaction\_time column.

CASE Statement:

WHEN HOUR(transaction\_time) >= 5 AND HOUR(transaction\_time) < 12 THEN 'Morning': If the hour is between 5 AM and before 12 PM, categorize it as 'Morning'.

WHEN HOUR(transaction\_time) >= 12 AND HOUR(transaction\_time) < 17 THEN 'Afternoon': If the hour is between 12 PM and before 5 PM, categorize it as 'Afternoon'.

WHEN HOUR(transaction\_time) >= 17 AND HOUR(transaction\_time) < 21 THEN 'Evening': If the hour is between 5 PM and before 9 PM, categorize it as 'Evening'.

ELSE 'Night': If none of the above conditions are met, categorize it as 'Night'.

--We have two diff columns like date and time. we want weekday from tha database.

so, Combine Date and Time Columns using concat function.

If your table has separate Date and Time columns, you can combine them into a datetime format and then extract the day of the week:

Explanation:

CONCAT(Date, ' ', Time): Combines the Date and Time columns into a single string in the format DD-MM-YYYY HH:MM:SS.

STR\_TO\_DATE(..., '%d-%m-%y %H:%i:%s'): Converts the combined string into a MySQL datetime format.

DAYNAME(): Extracts the day of the week from the datetime value

UPDATE WalmartSalesdata

SET week\_day = DAYNAME(STR\_TO\_DATE(CONCAT(Date, ' ', Time), '%d-%m-%Y %H:%i:%s'));

--Convert the Date Format Using STR\_TO\_DATE()

--You can use the STR\_TO\_DATE() function to convert the string date format into a recognized MySQL date format before using the MONTHNAME() function.

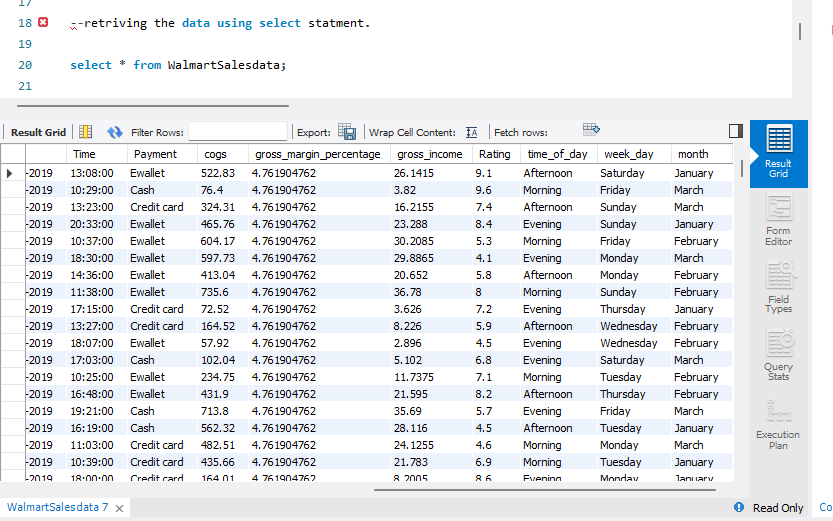
--Explanation:

STR\_TO\_DATE(Date, '%d-%m-%Y'): This converts the string Date in the format DD-MM-YYYY to a MySQL date format.

MONTHNAME(): This then extracts the month name from the converted date.

UPDATE WalmartSalesdata

SET month = MONTHNAME(STR\_TO\_DATE(Date, '%d-%m-%Y'));



--Q7. Which time of the day do customers give most ratings?

SELECT time\_of\_day,

count(rating) AS most\_rating

FROM WalmartSalesdata

GROUP BY time\_of\_day

ORDER BY most\_rating DESC;

Most ratings are done in the Afternoon.

This can be as a result that customers patronize the store more in the Afternoon.

A screenshot of a computer

Description automatically generated

--Q8. Which time of the day do customers give most ratings per branch?

SELECT branch, time\_of\_day, COUNT(\*) AS rating\_count

FROM WalmartSalesdata GROUP BY branch, time\_of\_day ORDER BY branch, rating\_count DESC;

Most of the ratings come in the afternoon time of the day with branch A.

A screenshot of a computer

Description automatically generated

--Q9. Which day fo the week has the best avg ratings?

SELECT week\_day, AVG(rating) AS Best\_Avg\_rating

FROM WalmartSalesdata GROUP BY week\_day ORDER BY Best\_Avg\_rating DESC;

Monday, Friday, and Sunday are the days of the week with the best average rating.

A screenshot of a computer

Description automatically generated

--Q10. Which day of the week has the best average ratings per branch?

SELECT week\_day, branch, AVG(rating) AS Best\_Avg\_rating\_pb

FROM WalmartSalesdata GROUP BY week\_day, branch ORDER BY Best\_Avg\_rating\_pb DESC, branch;

--pb stands form per branch

A screenshot of a computer

Description automatically generated