Walmart Sales Analysis

Walmart Sales Analysis

The aim of this project is to explore the Walmart Sales data to understand top performing branches and products, sales trend of different products, customer behaviour. The dataset given is of 45 Walmart stores located in different regions and each store contains many departments. 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 to what extent.

So, exploratory analysis is performed based on the dataset provided to answer questions of stakeholders about products, sales and customers.

Data Source:

The dataset is obtained from Kaggle Walmart Sales Forecasting Competition:

[https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting]

Dataset Description:

Column Name	Column Description
invoice_id	ID of the invoice for the sales made
branch	Name of the branch
city	Location of the branch
customer type	Type of customers who made purchase
gender	Gender of customers who made purchase
product_line	Product type of the product sold
unit_price	Price of each product
quantity	Quantity of the product sold
VAT	Amount of tax on purchase
total	Total cost of the purchase
date	Date on which the purchase was made
time	Time at which the purchase was made
payment_method	Method using which payment was done
cogs	Cost Of Goods Sold
gross_margin_percentage	Gross margin percentage
gross_income	Gross Income
rating	Rating

Tools Used:

- Microsoft SQL Server - For exploratory data analysis.

Steps Taken to perform Analysis:

Now let's look at the steps that I took to perform the analysis:

1. Data Wrangling:

Did inspection of data to make sure 'NULL' values and missing values are detected.

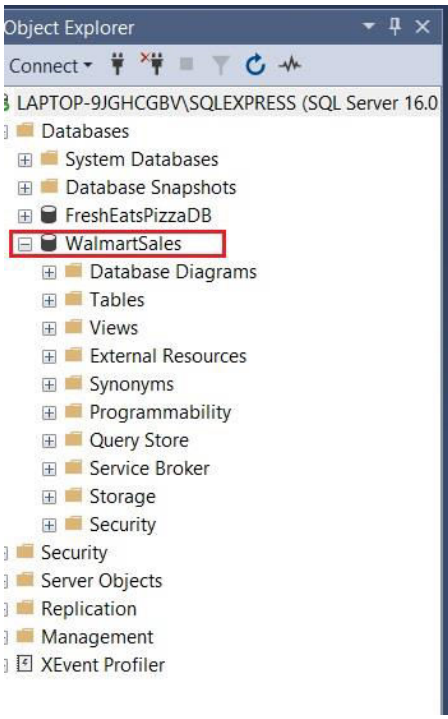
2. Build a Database.

WalmartQueries.sq...SAGAR SINGH (60))

```
-- Creating a Database if it does not exist in the system databases.
-----------------------------------------------------------------------
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name='WalmartSales')
  BEGIN
    CREATE DATABASE WalmartSales;
  END;
  GO

USE WalmartSales;
```

3. Create a table and insert data.

There are no null values in our database as while creating the tables each field was set to **NOT NULL** and hence null values got filtered out.



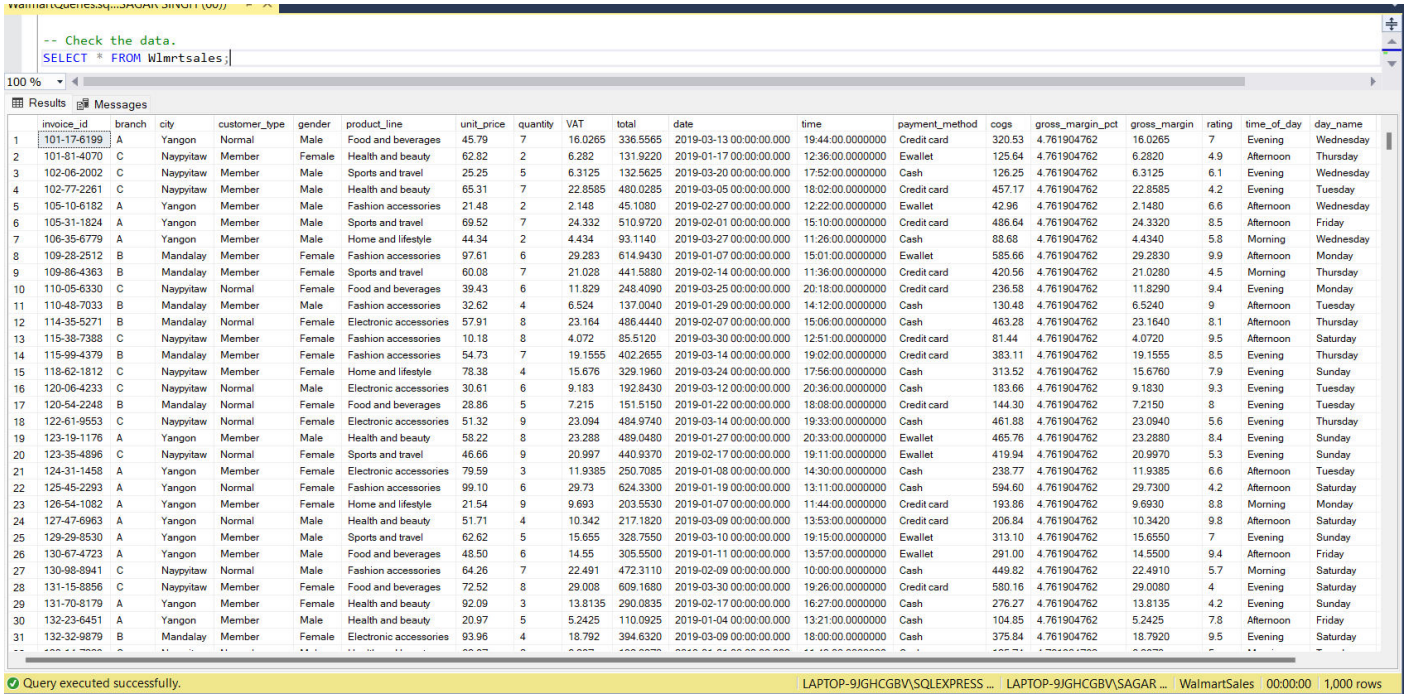
WalmartQueries.sq...SAGAR SINGH (60))

```
-- Creating a table in the database if it does not exist in the database.
-----------------------------------------------------------------------
IF OBJECT_ID(N'Wlmrtsales', N'U') IS NULL
BEGIN
 CREATE TABLE Wlmrtsales (
    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(15) NOT NULL,
    product_line VARCHAR(100) NOT NULL,
    unit_price DECIMAL(10, 2) NOT NULL,
    quantity INT NOT NULL,
    VAT FLOAT NOT NULL,
    total DECIMAL(12, 4) NOT NULL,
    date DATETIME NOT NULL,
    time TIME NOT NULL,
    payment_method VARCHAR(15) NOT NULL,
    cogs DECIMAL(10, 2) NOT NULL,
    gross_margin_pct FLOAT NOT NULL,
    gross_margin DECIMAL(12, 4) NOT NULL,
    rating FLOAT);
END;
```

WalmartQueries.sq...SAGAR SINGH (60))*

```
-- Bulk insert into the above table
---------------------------------------
BULK INSERT Wlmrtsales
FROM 'D:\DATA ANALYTICS\Projects\Walmart Sales Analysis\WalmartSalesData.csv'
WITH (FORMAT = 'CSV'
      , FIRSTROW = 2
      , FIELDTERMINATOR = ','
      , ROWTERMINATOR = '0x0a');
```

Check the imported data.



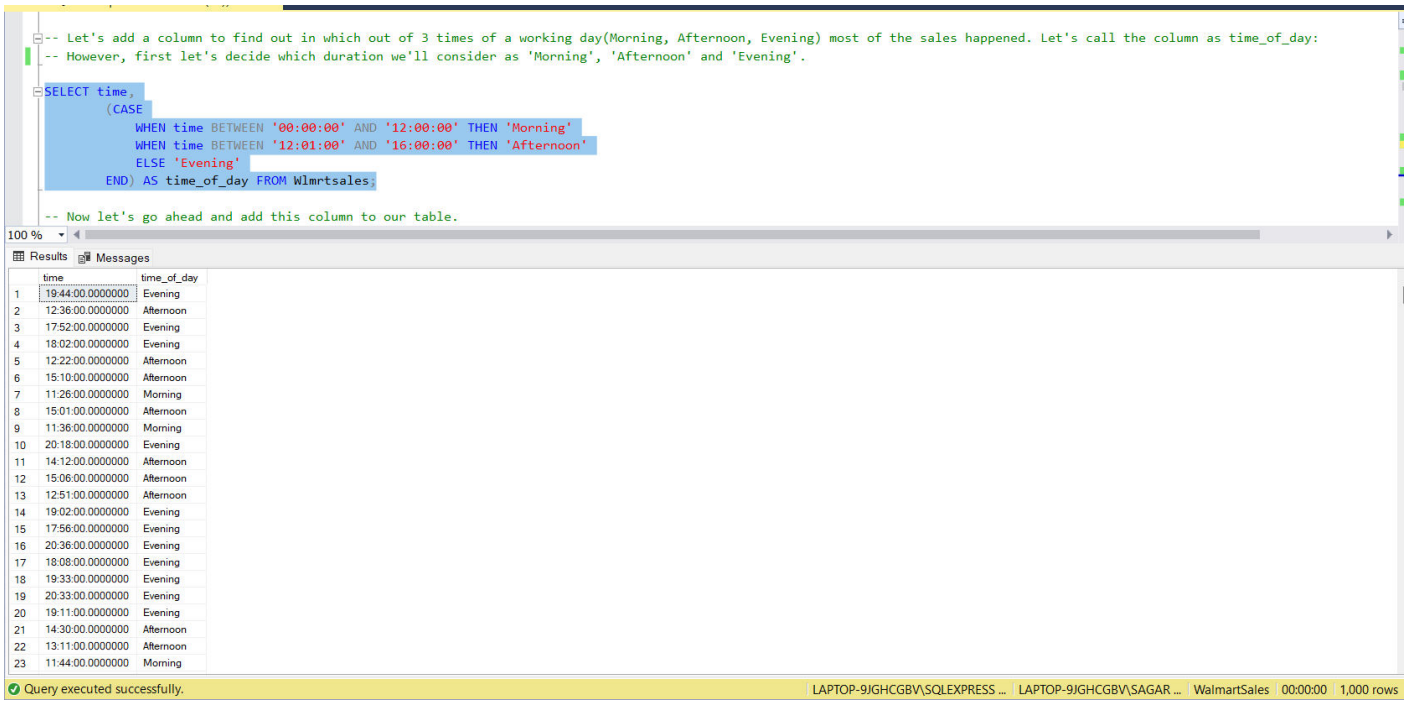
```
-- Check the data.
SELECT * FROM Wlmrtsales;
```

	invoice_id	branch	city	customer_type	gender	product_line	unit_price	quantity	VAT	total	date	time	payment_method	cogs	gross_margin_pct	gross_margin	rating	time_of_day	day_name
1	101-17-6199	A	Yangon	Normal	Male	Food and beverages	45.79	7	16.0265	336.5565	2019-03-13 00:00:00.000	19:44:00.0000000	Credit card	320.53	4.761904762	16.0265	7	Evening	Wednesday
2	101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	62.82	2	6.282	131.9220	2019-01-17 00:00:00.000	12:36:00.0000000	Ewallet	125.64	4.761904762	6.2820	4.9	Afternoon	Thursday
3	102-06-2002	C	Naypyitaw	Member	Male	Sports and travel	25.25	5	6.3125	132.5625	2019-03-20 00:00:00.000	17:52:00.0000000	Cash	126.25	4.761904762	6.3125	6.1	Evening	Wednesday
4	102-77-2261	C	Naypyitaw	Member	Male	Health and beauty	65.31	7	22.8585	480.0285	2019-03-05 00:00:00.000	18:02:00.0000000	Credit card	457.17	4.761904762	22.8585	4.2	Evening	Tuesday
5	105-10-6182	A	Yangon	Member	Male	Fashion accessories	21.48	2	2.148	45.1080	2019-02-27 00:00:00.000	12:22:00.0000000	Ewallet	42.96	4.761904762	2.1480	6.6	Afternoon	Wednesday
6	105-31-1824	A	Yangon	Member	Male	Sports and travel	69.52	7	24.332	510.9720	2019-02-01 00:00:00.000	15:10:00.0000000	Credit card	486.64	4.761904762	24.3320	8.5	Afternoon	Friday
7	106-35-6779	A	Yangon	Member	Male	Home and lifestyle	44.34	2	4.434	93.1140	2019-03-27 00:00:00.000	11:26:00.0000000	Cash	88.68	4.761904762	4.4340	5.8	Morning	Wednesday
8	109-28-2512	B	Mandalay	Member	Female	Fashion accessories	97.61	6	29.283	614.9430	2019-01-07 00:00:00.000	15:01:00.0000000	Ewallet	585.66	4.761904762	29.2830	9.9	Afternoon	Monday
9	109-86-4363	B	Mandalay	Member	Female	Sports and travel	60.08	7	21.028	441.5880	2019-02-14 00:00:00.000	11:36:00.0000000	Credit card	420.56	4.761904762	21.0280	4.5	Morning	Thursday
10	110-05-6330	C	Naypyitaw	Normal	Female	Food and beverages	39.43	6	11.829	248.4090	2019-03-25 00:00:00.000	20:18:00.0000000	Credit card	236.58	4.761904762	11.8290	9.4	Evening	Monday
11	110-48-7033	B	Mandalay	Member	Male	Fashion accessories	32.62	4	6.524	137.0040	2019-01-29 00:00:00.000	14:12:00.0000000	Cash	130.48	4.761904762	6.5240	9	Afternoon	Tuesday
12	114-35-5271	B	Mandalay	Normal	Female	Electronic accessories	57.91	8	23.164	486.4440	2019-02-07 00:00:00.000	15:06:00.0000000	Cash	463.28	4.761904762	23.1640	8.1	Afternoon	Thursday
13	115-38-7388	C	Naypyitaw	Member	Female	Fashion accessories	10.18	8	4.072	85.5120	2019-03-30 00:00:00.000	12:51:00.0000000	Credit card	81.44	4.761904762	4.0720	9.5	Afternoon	Saturday
14	115-99-4379	B	Mandalay	Member	Female	Fashion accessories	54.73	7	19.1555	402.2655	2019-03-14 00:00:00.000	19:02:00.0000000	Credit card	383.11	4.761904762	19.1555	8.5	Evening	Thursday
15	118-62-1812	C	Naypyitaw	Member	Female	Home and lifestyle	78.38	4	15.676	329.1960	2019-03-24 00:00:00.000	17:56:00.0000000	Cash	313.52	4.761904762	15.6760	7.9	Evening	Sunday
16	120-06-4233	C	Naypyitaw	Normal	Male	Electronic accessories	30.61	6	9.183	192.8430	2019-03-12 00:00:00.000	20:36:00.0000000	Cash	183.66	4.761904762	9.1830	9.3	Evening	Tuesday
17	120-54-2248	B	Mandalay	Normal	Female	Food and beverages	28.86	5	7.215	151.5150	2019-01-22 00:00:00.000	18:08:00.0000000	Credit card	144.30	4.761904762	7.2150	8	Evening	Tuesday
18	122-61-9553	C	Naypyitaw	Normal	Female	Electronic accessories	51.32	9	23.094	484.9740	2019-03-14 00:00:00.000	19:33:00.0000000	Cash	461.88	4.761904762	23.0940	5.6	Evening	Thursday
19	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.288	489.0480	2019-01-27 00:00:00.000	20:33:00.0000000	Ewallet	465.76	4.761904762	23.2880	8.4	Evening	Sunday
20	123-35-4896	C	Naypyitaw	Normal	Female	Sports and travel	46.66	9	20.997	440.9370	2019-02-17 00:00:00.000	19:11:00.0000000	Ewallet	419.94	4.761904762	20.9970	5.3	Evening	Sunday
21	124-31-1458	A	Yangon	Member	Female	Electronic accessories	79.59	3	11.9385	250.7085	2019-01-08 00:00:00.000	14:30:00.0000000	Cash	238.77	4.761904762	11.9385	6.6	Afternoon	Tuesday
22	125-45-2293	A	Yangon	Normal	Female	Fashion accessories	99.10	6	29.73	624.3300	2019-01-19 00:00:00.000	13:11:00.0000000	Cash	594.60	4.761904762	29.7300	4.2	Afternoon	Saturday
23	126-54-1082	A	Yangon	Member	Female	Home and lifestyle	21.54	9	9.693	203.5530	2019-01-07 00:00:00.000	11:44:00.0000000	Credit card	193.86	4.761904762	9.6930	8.8	Morning	Monday
24	127-47-6963	A	Yangon	Normal	Male	Health and beauty	51.71	4	10.342	217.1820	2019-03-09 00:00:00.000	13:53:00.0000000	Credit card	206.84	4.761904762	10.3420	9.8	Afternoon	Saturday
25	129-29-8530	A	Yangon	Member	Male	Sports and travel	62.62	5	15.655	328.7550	2019-03-10 00:00:00.000	19:15:00.0000000	Ewallet	313.10	4.761904762	15.6550	7	Evening	Sunday
26	130-67-4723	A	Yangon	Member	Male	Food and beverages	48.50	6	14.55	305.5500	2019-01-11 00:00:00.000	13:57:00.0000000	Ewallet	291.00	4.761904762	14.5500	9.4	Afternoon	Friday
27	130-98-8941	C	Naypyitaw	Normal	Male	Fashion accessories	64.26	7	22.491	472.3110	2019-02-09 00:00:00.000	10:00:00.0000000	Cash	449.82	4.761904762	22.4910	5.7	Morning	Saturday
28	131-15-8856	C	Naypyitaw	Member	Female	Food and beverages	72.52	8	29.008	609.1680	2019-03-30 00:00:00.000	19:26:00.0000000	Credit card	580.16	4.761904762	29.0080	4	Evening	Saturday
29	131-70-8179	A	Yangon	Member	Female	Health and beauty	92.09	3	13.8135	290.0835	2019-02-17 00:00:00.000	16:27:00.0000000	Cash	276.27	4.761904762	13.8135	4.2	Evening	Sunday
30	132-23-6451	A	Yangon	Member	Male	Health and beauty	20.97	5	5.2425	110.0925	2019-01-04 00:00:00.000	13:21:00.0000000	Cash	104.85	4.761904762	5.2425	7.8	Afternoon	Friday
31	132-32-9879	B	Mandalay	Member	Female	Electronic accessories	93.96	4	18.792	394.6320	2019-03-09 00:00:00.000	18:00:00.0000000	Cash	375.84	4.761904762	18.7920	9.5	Evening	Saturday

Let’s add some columns using the existing ones which are going to be helpful while answering questions:

1. Let's add a column to find out in which out of 3 times of a working day (Morning, Afternoon, Evening) most of the sales happened. Let's call the column as time_of_day:

However, first let's decide which duration we'll consider as 'Morning', 'Afternoon' and 'Evening'.



```
-- Let's add a column to find out in which out of 3 times of a working day(Morning, Afternoon, Evening) most of the sales happened. Let's call the column as time_of_day:
-- However, first let's decide which duration we'll consider as 'Morning', 'Afternoon' and 'Evening'.

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 Wlmrtsales;

-- Now let's go ahead and add this column to our table.
```

	time	time_of_day
1	19:44:00.0000000	Evening
2	12:36:00.0000000	Afternoon
3	17:52:00.0000000	Evening
4	18:02:00.0000000	Evening
5	12:22:00.0000000	Afternoon
6	15:10:00.0000000	Afternoon
7	11:26:00.0000000	Morning
8	15:01:00.0000000	Afternoon
9	11:36:00.0000000	Morning
10	20:18:00.0000000	Evening
11	14:12:00.0000000	Afternoon
12	15:06:00.0000000	Afternoon
13	12:51:00.0000000	Afternoon
14	19:02:00.0000000	Evening
15	17:56:00.0000000	Evening
16	20:36:00.0000000	Evening
17	18:08:00.0000000	Evening
18	19:33:00.0000000	Evening
19	20:33:00.0000000	Evening
20	19:11:00.0000000	Evening
21	14:30:00.0000000	Afternoon
22	13:11:00.0000000	Afternoon
23	11:44:00.0000000	Morning

Now let's go ahead and add this column to our table. If we check, this column is empty for now and will show NULL values. So, let's insert some data into this column.

```sql
-- Now let's go ahead and add this column to our table.
ALTER TABLE Wlmrtsales ADD time_of_day VARCHAR(20);

-- If we check, this column is empty for now and will show NULL values.
SELECT time_of_day FROM Wlmrtsales;

--Let's insert some data into this column.
UPDATE Wlmrtsales 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);
```

Now let's check the same column again. We'll see Morning, Afternoon and Evening filled respectively according to hours.

```sql
-- Now let's check the same column again. We'll see Morning, Afternoon and Evening filled respectively according to hours.
SELECT time, time_of_day FROM Wlmrtsales;
```

100 %

Results | Messages

	time	time_of_day
1	19:44:00.0000000	Evening
2	12:36:00.0000000	Afternoon
3	17:52:00.0000000	Evening
4	18:02:00.0000000	Evening
5	12:22:00.0000000	Afternoon
6	15:10:00.0000000	Afternoon
7	11:26:00.0000000	Morning
8	15:01:00.0000000	Afternoon
9	11:36:00.0000000	Morning
10	20:18:00.0000000	Evening
11	14:12:00.0000000	Afternoon
12	15:06:00.0000000	Afternoon
13	12:51:00.0000000	Afternoon
14	19:02:00.0000000	Evening
15	17:56:00.0000000	Evening
16	20:36:00.0000000	Evening
17	18:08:00.0000000	Evening
18	19:33:00.0000000	Evening
19	20:33:00.0000000	Evening
20	19:11:00.0000000	Evening
21	14:30:00.0000000	Afternoon
22	13:11:00.0000000	Afternoon
23	11:44:00.0000000	Morning
24	13:53:00.0000000	Afternoon
25	19:15:00.0000000	Evening
26	13:57:00.0000000	Afternoon
27	10:00:00.0000000	Morning
28	19:26:00.0000000	Evening
29	16:27:00.0000000	Evening
30	13:21:00.0000000	Afternoon
31	18:00:00.0000000	Evening
32	11:43:00.0000000	Morning
33	18:04:00.0000000	Evening

Query executed successfully. LAPTOP-9JGHCGBV\SQLEXPRESS ... LAPTO

2. Let's add columns for day names (Mon, Tue, Wed, Thurs, Fri) and month names (Jan, Feb, Mar) so that we can also find how was sales on which day and in which month. We have to follow the same steps as in step 1.

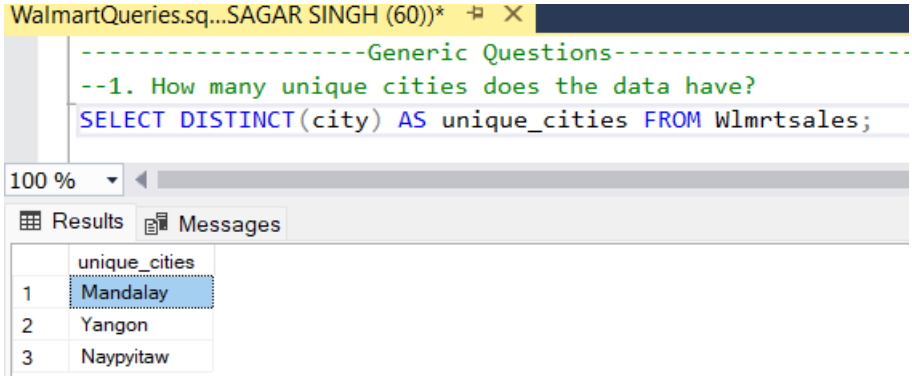
```sql
-- Let's add columns for day names (Mon, Tue, Wed, Thur, Fri) and month names (Jan, Feb, Mar) so that we can also find how was sales on which day and in which month.
-- day names
SELECT date,
DATENAME(weekday, date) AS day_name FROM Wlmrtsales;

ALTER TABLE Wlmrtsales ADD day_name VARCHAR(10);

UPDATE Wlmrtsales SET day_name = (DATENAME(weekday, date));

-- month names
SELECT date,
DATENAME(month, date) AS month_name FROM Wlmrtsales;

ALTER TABLE Wlmrtsales ADD month_name VARCHAR(20);

UPDATE Wlmrtsales SET month_name = (DATENAME(month, date));
```

Business Questions:

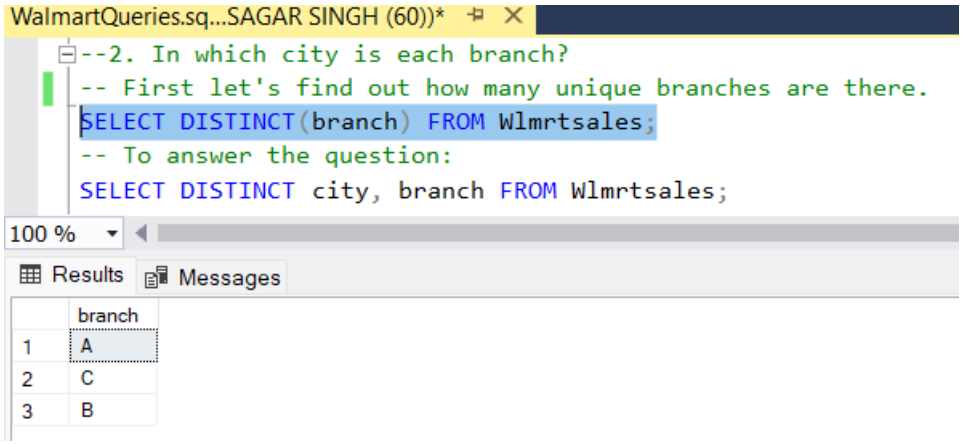
It is now time to look at stakeholders' questions and find answers:

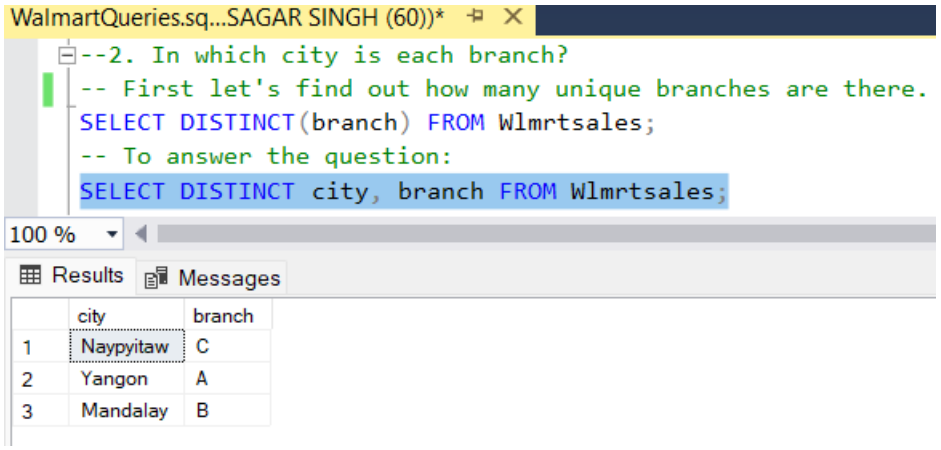
Generic Questions:

1. How many unique cities does the data have?



2. In which city is each branch?





Product Related Questions:

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

```
----------Product Related Questions----------
--1. How many unique product lines does the data have?
SELECT DISTINCT product_line FROM Wlmrtsales; -- This will show different product lines
SELECT COUNT(DISTINCT product_line) AS unique_products FROM Wlmrtsales; -- This will show the number of different product lines.
```

	product_line
1	Fashion accessories
2	Home and lifestyle
3	Electronic accessories
4	Health and beauty
5	Sports and travel
6	Food and beverages

```
----------Product Related Questions----------
--1. How many unique product lines does the data have?
SELECT DISTINCT product_line FROM Wlmrtsales; -- This will show different product lines
SELECT COUNT(DISTINCT product_line) AS unique_products FROM Wlmrtsales; -- This will show the number of different product lines.
```

	unique_products
1	6

2. What is the most common payment method?

```
--2. What is the most common payment method?
-- Let's first see what are the payment method options available in the data.
SELECT DISTINCT payment_method FROM Wlmrtsales;
-- To answer the question
SELECT payment_method, COUNT(payment_method) AS max_count FROM Wlmrtsales GROUP BY payment_method ORDER BY max_count DESC;
```

	payment_method
1	Credit card
2	Cash
3	Ewallet

```
--2. What is the most common payment method?
-- Let's first see what are the payment method options available in the data.
SELECT DISTINCT payment_method FROM Wlmrtsales;
-- To answer the question
SELECT payment_method, COUNT(payment_method) AS max_count FROM Wlmrtsales GROUP BY payment_method ORDER BY max_count DESC;
```

	payment_method	max_count
1	Ewallet	345
2	Cash	344
3	Credit card	311

Hence, 'Ewallet' is the most common payment method. It is used maximum times.

3. What is the most selling product line?

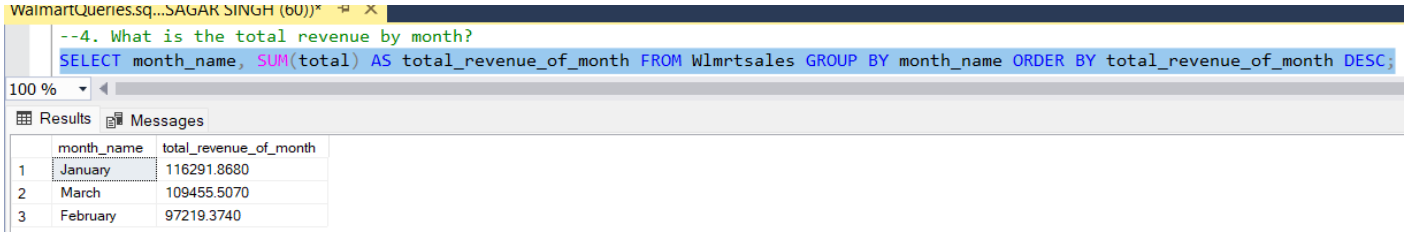
```
--3. What is the most selling product line?
SELECT product_line, COUNT(product_line) AS max_product_count FROM Wlmrtsales GROUP BY product_line ORDER BY max_product_count DESC;
```

	product_line	max_product_count
1	Fashion accessories	178
2	Food and beverages	174
3	Electronic accessories	170
4	Sports and travel	166
5	Home and lifestyle	160
6	Health and beauty	152

Fashion accessories is the most selling product line.

4. What is the total revenue by month?

If we look at the beginning of the document in the data columns description, then 'total' column is 'Total cost of the purchase' which can give us revenue. So, we can use this column to get our answer.



```
--4. What is the total revenue by month?
SELECT month_name, SUM(total) AS total_revenue_of_month FROM Wlmrtsales GROUP BY month_name ORDER BY total_revenue_of_month DESC;
```

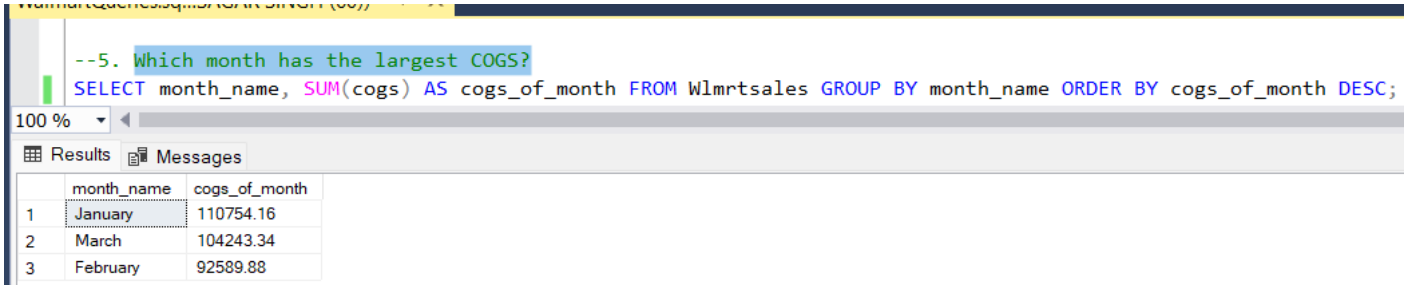
100 %

Results | Messages

	month_name	total_revenue_of_month
1	January	116291.8680
2	March	109455.5070
3	February	97219.3740

5. Which month has the largest COGS?

We have a cogs column in our data which we can use to answer this question.



```
--5. Which month has the largest COGS?
SELECT month_name, SUM(cogs) AS cogs_of_month FROM Wlmrtsales GROUP BY month_name ORDER BY cogs_of_month DESC;
```

100 %

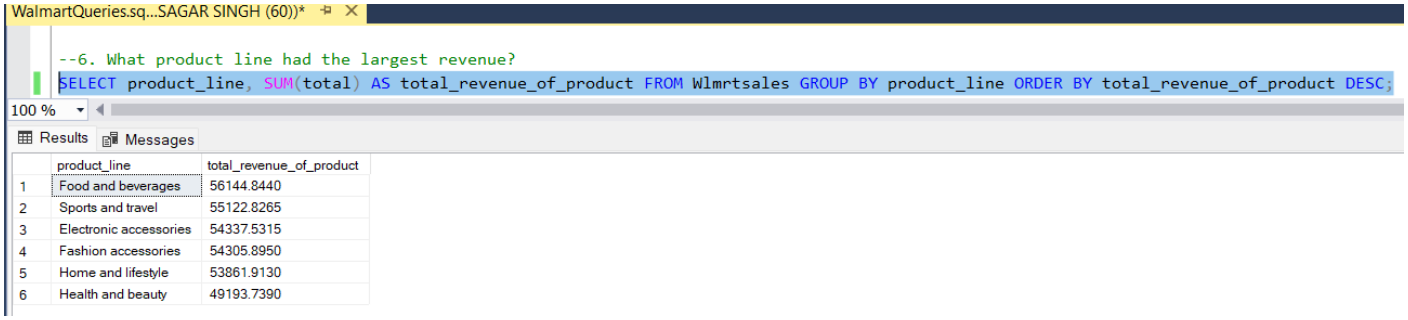
Results | Messages

	month_name	cogs_of_month
1	January	110754.16
2	March	104243.34
3	February	92589.88

January has the largest 'Cost of Goods Sold'.

6. What product line had the largest revenue?

We can use 'total' column for revenue.



```
--6. What product line had the largest revenue?
SELECT product_line, SUM(total) AS total_revenue_of_product FROM Wlmrtsales GROUP BY product_line ORDER BY total_revenue_of_product DESC;
```

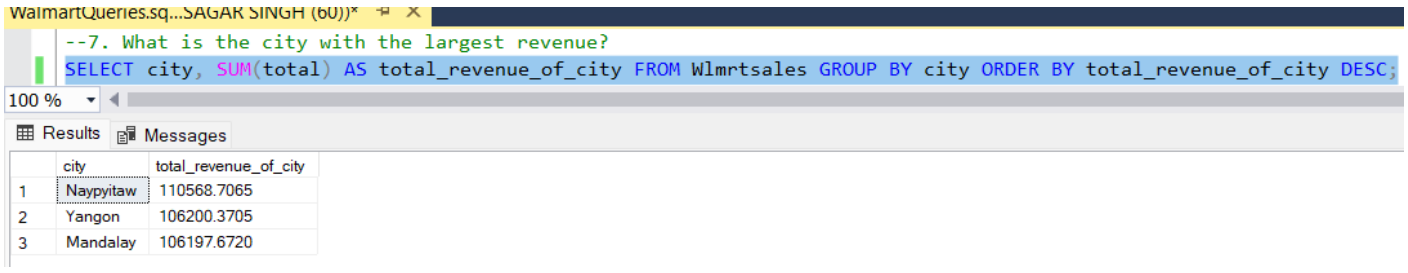
100 %

Results | Messages

	product_line	total_revenue_of_product
1	Food and beverages	56144.8440
2	Sports and travel	55122.8265
3	Electronic accessories	54337.5315
4	Fashion accessories	54305.8950
5	Home and lifestyle	53861.9130
6	Health and beauty	49193.7390

Foods and beverages had the largest revenue.

7. What is the city with the largest revenue?



```
--7. What is the city with the largest revenue?
SELECT city, SUM(total) AS total_revenue_of_city FROM Wlmrtsales GROUP BY city ORDER BY total_revenue_of_city DESC;
```

100 %

Results | Messages

	city	total_revenue_of_city
1	Naypyitaw	110568.7065
2	Yangon	106200.3705
3	Mandalay	106197.6720

Naypyitaw is the city with the largest revenue.

8. What product line had the largest VAT?

We already have a VAT (Value Added Tax) column in our dataset. So, we can use that:

```
--8. What product line had the largest VAT?
SELECT product_line, SUM(VAT) AS total_VAT_of_product FROM Wlmrtsales GROUP BY product_line ORDER BY total_VAT_of_product DESC;
```

	product_line	total_VAT_of_product
1	Food and beverages	2673.564
2	Sports and travel	2624.8965
3	Electronic accessories	2587.5015
4	Fashion accessories	2585.995
5	Home and lifestyle	2564.853
6	Health and beauty	2342.559

Food and beverages had the largest VAT.

9. Fetch each product line and add a column to those product line showing "Good", "Bad". Good if its sales is greater than the average sales.

```
--9. Fetch each product line and add a column to those product line showing "Good", "Bad". Good if its sales is greater than the average sales.
SELECT product_line, cogs,
                    CASE
                          WHEN cogs > (SELECT AVG(cogs) FROM Wlmrtsales) THEN 'Good'
                          ELSE 'Bad'
                     END AS Product_Status FROM Wlmrtsales;
```

	product_line	cogs	Product_Status
1	Food and beverages	320.53	Good
2	Health and beauty	125.64	Bad
3	Sports and travel	126.25	Bad
4	Health and beauty	457.17	Good
5	Fashion accessories	42.96	Bad
6	Sports and travel	486.64	Good
7	Home and lifestyle	88.68	Bad
8	Fashion accessories	585.66	Good
9	Sports and travel	420.56	Good
10	Food and beverages	236.58	Bad
11	Fashion accessories	130.48	Bad
12	Electronic accessories	463.28	Good
13	Fashion accessories	81.44	Bad
14	Fashion accessories	383.11	Good
15	Home and lifestyle	313.52	Good
16	Electronic accessories	183.66	Bad
17	Food and beverages	144.30	Bad
18	Electronic accessories	461.88	Good
19	Health and beauty	465.76	Good
20	Sports and travel	419.94	Good
21	Electronic accessories	238.77	Bad
22	Fashion accessories	594.60	Good
23	Home and lifestyle	193.86	Bad
24	Health and beauty	206.84	Bad
25	Sports and travel	313.10	Good
26	Food and beverages	291.00	Bad
27	Fashion accessories	449.82	Good
28	Food and beverages	580.16	Good
29	Health and beauty	276.27	Bad

Query executed successfully. LAPTOP-9JGHCGBV\SQLEXPRESS ... LAPTOP-9JGHCGBV

10. Which branches sold more products than average product sold?

```
--10. Which branches sold more products than average product sold?
SELECT branch, COUNT(quantity) AS Products_sold FROM Wlmrtsales GROUP BY branch HAVING COUNT(quantity) > AVG(quantity);
```

	branch	Products_sold
1	A	340
2	C	328
3	B	332

Hence, all three branches A, B and C sold more products than the average product sold.

11. What is the most common product line by gender?

```
--11. What is the most common product line by gender?
SELECT gender, product_line, COUNT(product_line) AS most_com_product FROM Wlmrtsales GROUP BY gender, product_line ORDER BY most_com_product DESC;
```

100 %

Results | Messages

	gender	product_line	most_com_product
1	Female	Fashion accessories	96
2	Female	Food and beverages	90
3	Female	Sports and travel	88
4	Male	Health and beauty	88
5	Male	Electronic accessories	86
6	Female	Electronic accessories	84
7	Male	Food and beverages	84
8	Male	Fashion accessories	82
9	Male	Home and lifestyle	81
10	Female	Home and lifestyle	79
11	Male	Sports and travel	78
12	Female	Health and beauty	64

So, for 'Female' the most common product line is 'Fashion accessories' and for 'Male' it is 'Health and beauty'.

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

WalmartQueries.sq...SAGAR SINGH (60))

```
--12. What is the average rating of each product line?
SELECT product_line, ROUND(AVG(rating), 2) AS avg_rating FROM Wlmrtsales GROUP BY product_line ORDER BY avg_rating DESC;
```

100 %

Results | Messages

	product_line	avg_rating
1	Food and beverages	7.11
2	Fashion accessories	7.03
3	Health and beauty	7
4	Sports and travel	6.92
5	Electronic accessories	6.92
6	Home and lifestyle	6.84

Hence, average rating for 'Food and beverages' is 7.11, 'Fashion accessories' is 7.03 and so on.

Sales Related Questions:

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

```
----------Sales Related Questions----------
--1. Number of sales made in each time of the day per weekday.
SELECT day_name, time_of_day, COUNT(quantity) AS num_of_sales FROM Wlmrtsales GROUP BY day_name, time_of_day ORDER BY num_of_sales DESC;
```

100 %

Results | Messages

	day_name	time_of_day	num_of_sales
1	Saturday	Evening	81
2	Tuesday	Evening	69
3	Wednesday	Afternoon	61
4	Wednesday	Evening	60
5	Friday	Afternoon	58
6	Sunday	Evening	58
7	Monday	Evening	56
8	Thursday	Evening	56
9	Saturday	Afternoon	55
10	Sunday	Afternoon	53
11	Tuesday	Afternoon	53
12	Friday	Evening	52
13	Thursday	Afternoon	49
14	Monday	Afternoon	48
15	Tuesday	Morning	36
16	Thursday	Morning	33
17	Friday	Morning	29
18	Saturday	Morning	28
19	Wednesday	Morning	22
20	Sunday	Morning	22
21	Monday	Morning	21

2. Which of the customer type brings the most revenue?

Again, when it comes to revenue, we can use 'total' column from our data. However, let us first see different customer types in our data.

```
--2. Which of the customer type brings the most revenue?
--Let us first see the different customer types.
SELECT DISTINCT customer_type FROM Wlmrtsales;
--Now answer to the question.
SELECT customer_type, SUM(total) AS revenue FROM Wlmrtsales GROUP BY customer_type ORDER BY revenue DESC;
```

100 %

Results | Messages

	customer_type
1	Member
2	Normal

Now, let's answer the question:

```
--2. Which of the customer type brings the most revenue?
--Let us first see the different customer types.
SELECT DISTINCT customer_type FROM Wlmrtsales;
--Now answer to the question.
SELECT customer_type, SUM(total) AS revenue FROM Wlmrtsales GROUP BY customer_type ORDER BY revenue DESC;
```

100 %

Results | Messages

	customer_type	revenue
1	Member	164223.4440
2	Normal	158743.3050

Member type customer brings the most revenue.

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

```
--3. Which city has the largest tax / VAT (**Value Added Tax**)?
SELECT city, SUM(VAT) AS tax FROM Wlmrtsales GROUP BY city ORDER BY tax DESC;
```

100 %

Results | Messages

	city	tax
1	Naypyitaw	5265.1765
2	Yangon	5057.1605
3	Mandalay	5057.03200000001

Naypyitaw has the tax / VAT.

4. Which customer type pays the most in VAT?

```
--4. Which customer type pays the most in VAT?
SELECT customer_type, SUM(VAT) AS V_A_T FROM Wlmrtsales GROUP BY customer_type ORDER BY V_A_T DESC;
```

100 %

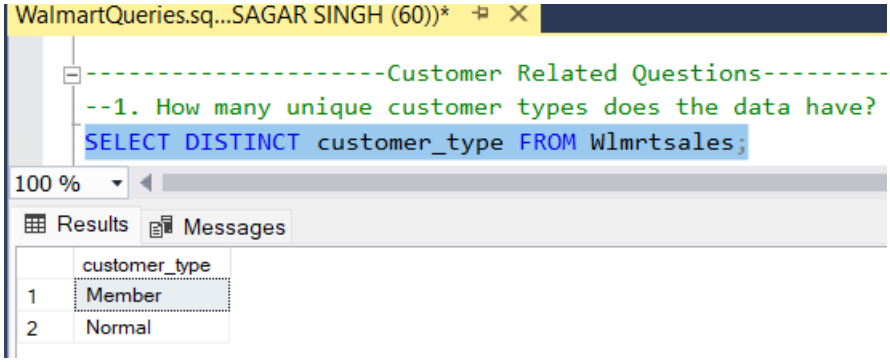
Results | Messages

	customer_type	V_A_T
1	Member	7820.164
2	Normal	7559.20500000001

Customer Related Questions:

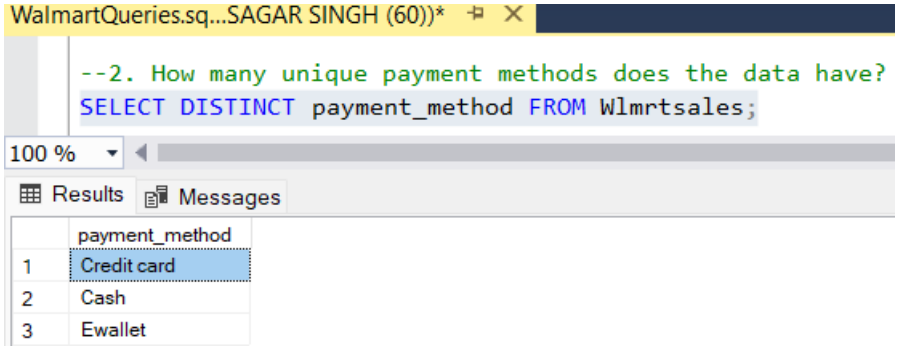
1. How many unique customer types does the data have?

I think we have already found answer to this question while answering the second question of Sales section.

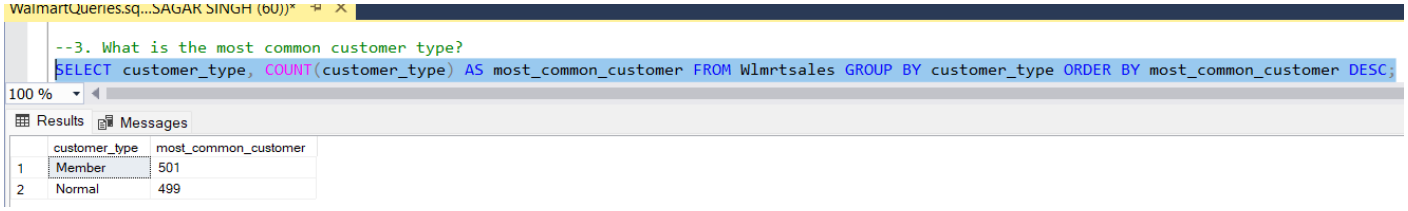


The answer is 'Member' and 'Normal'.

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

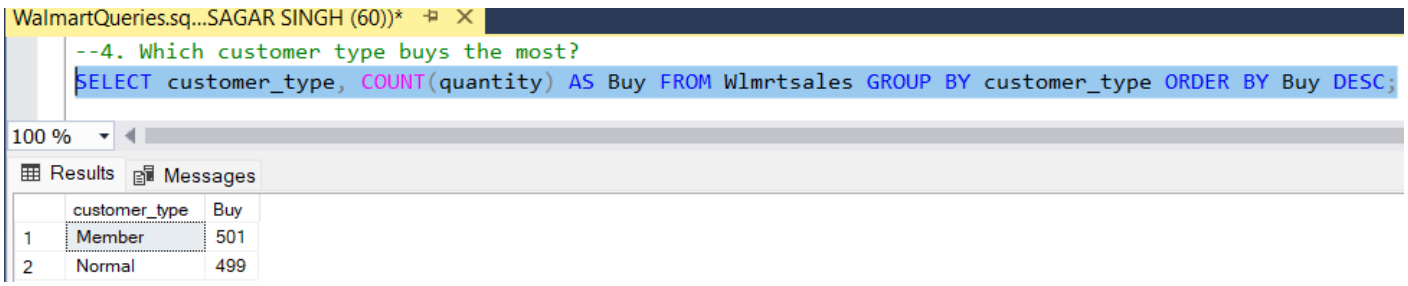


3. What is the most common customer type?



Member customer type is the most common.

4. Which customer type buys the most?



Member customer type buys the most.

5. What is the gender of most of the customers?

```
--5. What is the gender of most of the customers?
SELECT gender, COUNT(gender) AS gender_count FROM Wlmrtsales GROUP BY gender ORDER BY gender_count DESC;
```

	gender	gender_count
1	Female	501
2	Male	499

Female

6. What is the gender distribution per branch?

```
--6. What is the gender distribution per branch?
SELECT branch, gender, COUNT(gender) AS gender_distribution FROM Wlmrtsales GROUP BY branch, gender ORDER BY gender_distribution DESC;
```

	branch	gender	gender_distribution
1	A	Male	179
2	C	Female	178
3	B	Male	170
4	B	Female	162
5	A	Female	161
6	C	Male	150

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

```
--7. Which time of the day do customers give most ratings?
SELECT time_of_day, COUNT(rating) AS number_of_ratings FROM Wlmrtsales GROUP BY time_of_day ORDER BY number_of_ratings DESC;
```

	time_of_day	number_of_ratings
1	Evening	432
2	Afternoon	377
3	Morning	191

Customers mostly give ratings in the ‘Evening’.

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

```
--8. Which time of the day do customers give most ratings per branch?
SELECT time_of_day, branch, COUNT(rating) AS number_of_ratings FROM Wlmrtsales GROUP BY time_of_day, branch ORDER BY number_of_ratings DESC;
```

	time_of_day	branch	number_of_ratings
1	Evening	B	148
2	Evening	C	143
3	Evening	A	141
4	Afternoon	A	126
5	Afternoon	C	126
6	Afternoon	B	125
7	Morning	A	73
8	Morning	C	59
9	Morning	B	59

In the ‘Evening’ time customers give most ratings per branch.

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

WalmartQueries.sq...SAGAR SINGH (60))*

```
--9. Which day of the week has the best avg ratings?
SELECT day_name, AVG(rating) AS avg_rating FROM Wlmrtsales GROUP BY day_name ORDER BY avg_rating DESC;
```

100 %

Results | Messages

	day_name	avg_rating
1	Monday	7.1536
2	Friday	7.07625899280576
3	Sunday	7.01127819548872
4	Tuesday	7.00316455696203
5	Saturday	6.90182926829268
6	Thursday	6.88985507246377
7	Wednesday	6.8055944055944

Monday.

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

WalmartQueries.sq...SAGAR SINGH (60))*

```
--10. Which day of the week has the best average ratings per branch?
SELECT day_name, branch, AVG(rating) AS avg_rating FROM Wlmrtsales GROUP BY day_name, branch ORDER BY avg_rating DESC;
```

100 %

Results | Messages

	day_name	branch	avg_rating
1	Monday	B	7.33589743589743
2	Friday	A	7.312
3	Friday	C	7.27894736842105
4	Saturday	C	7.22962962962963
5	Monday	A	7.09791666666667
6	Sunday	A	7.07884615384616
7	Wednesday	C	7.064
8	Tuesday	A	7.05882352941176
9	Monday	C	7.03684210526316
10	Sunday	C	7.02826086956522
11	Tuesday	B	7.00188679245283
12	Thursday	A	6.95869565217391
13	Tuesday	C	6.95185185185185
14	Thursday	C	6.95
15	Wednesday	A	6.91627906976744
16	Sunday	B	6.88857142857143
17	Thursday	B	6.75227272727273
18	Saturday	A	6.746
19	Saturday	B	6.73666666666667
20	Friday	B	6.69411764705883
21	Wednesday	B	6.452

Monday has the best average ratings per branch.

The SQL file with all the above-mentioned queries can be found on my GitHub repository (https://github.com/singhocean/Walmart-Sales-Analysis).