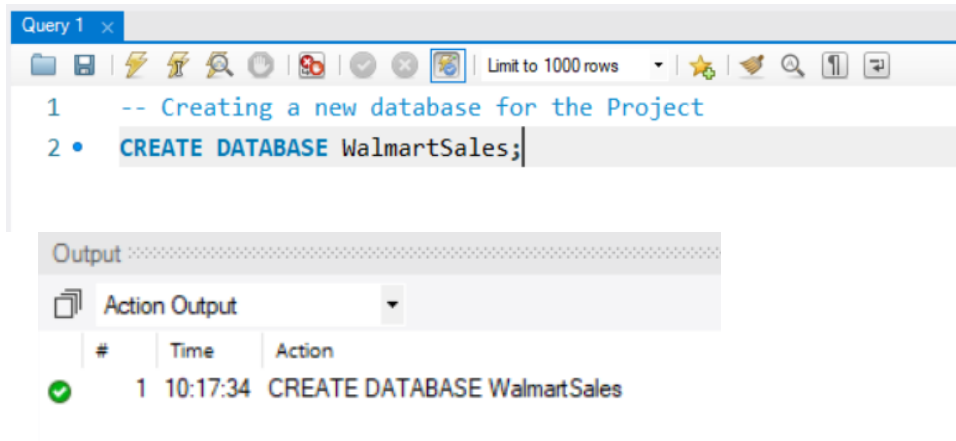


SQL Queries for Walmart Project

1. Creating a new Database WalmartSales for the Project



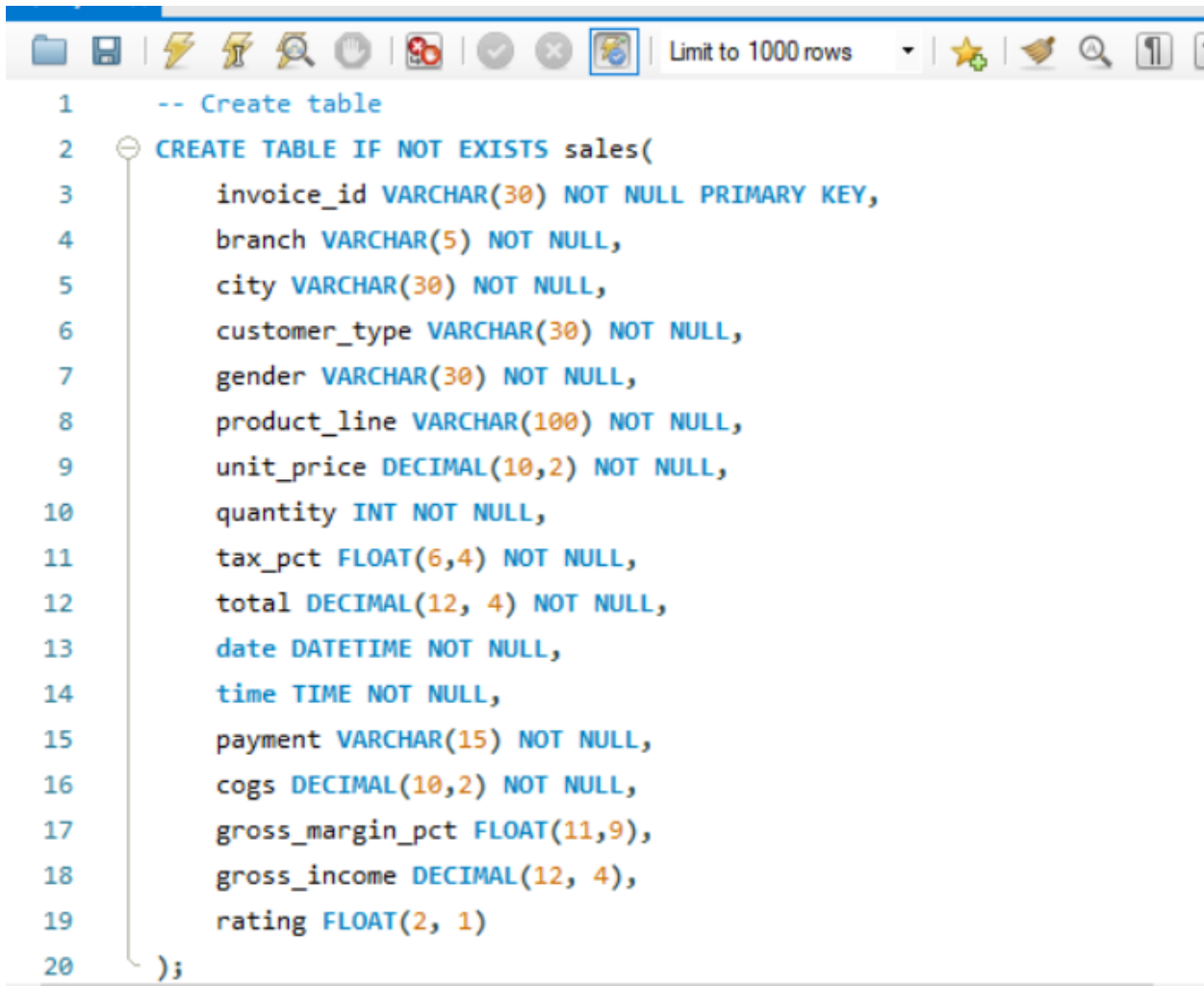
The screenshot shows a SQL query editor window titled "Query 1". The toolbar includes icons for file operations, execution, and a "Limit to 1000 rows" dropdown. The query text is as follows:

```
1 -- Creating a new database for the Project
2 • CREATE DATABASE WalmartSales;
```

Below the query editor is an "Output" section with a dropdown menu set to "Action Output". It displays a single action:

#	Time	Action
✓ 1	10:17:34	CREATE DATABASE WalmartSales

2. Creating new table sales and importing Data from Import Option directly into sales table



The screenshot shows a SQL query editor window with a toolbar similar to the first one. The query text is as follows:

```
1 -- Create table
2 CREATE TABLE IF NOT EXISTS sales(
3     invoice_id VARCHAR(30) NOT NULL PRIMARY KEY,
4     branch VARCHAR(5) NOT NULL,
5     city VARCHAR(30) NOT NULL,
6     customer_type VARCHAR(30) NOT NULL,
7     gender VARCHAR(30) NOT NULL,
8     product_line VARCHAR(100) NOT NULL,
9     unit_price DECIMAL(10,2) NOT NULL,
10    quantity INT NOT NULL,
11    tax_pct FLOAT(6,4) NOT NULL,
12    total DECIMAL(12, 4) NOT NULL,
13    date DATETIME NOT NULL,
14    time TIME NOT NULL,
15    payment VARCHAR(15) NOT NULL,
16    cogs DECIMAL(10,2) NOT NULL,
17    gross_margin_pct FLOAT(11,9),
18    gross_income DECIMAL(12, 4),
19    rating FLOAT(2, 1)
20 );
```

```

1  SELECT
2      *
3  FROM sales;
4

```

invoice_id	branch	city	customer_type	gender	product_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs
101-17-6199	A	Yangon	Normal	Male	Food and beverages	45.79	7	16.0265	336.5565	2019-03-13 00:00:00	19:44:00	Credit card	320.53
101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	62.82	2	6.2820	131.9220	2019-01-17 00:00:00	12:36:00	Ewallet	125.64
102-06-2002	C	Naypyitaw	Member	Male	Sports and travel	25.25	5	6.3125	132.5625	2019-03-20 00:00:00	17:52:00	Cash	126.25
102-77-2261	C	Naypyitaw	Member	Male	Health and beauty	65.31	7	22.8585	480.0285	2019-03-05 00:00:00	18:02:00	Credit card	457.17
105-10-6182	A	Yangon	Member	Male	Fashion accessories	21.48	2	2.1480	45.1080	2019-02-27 00:00:00	12:22:00	Ewallet	42.96
105-31-1824	A	Yangon	Member	Male	Sports and travel	69.52	7	24.3320	510.9720	2019-02-01 00:00:00	15:10:00	Credit card	486.64
106-35-6779	A	Yangon	Member	Male	Home and lifestyle	44.34	2	4.4340	93.1140	2019-03-27 00:00:00	11:26:00	Cash	88.68
109-28-2512	B	Mandalay	Member	Female	Fashion accessories	97.61	6	29.2830	614.9430	2019-01-07 00:00:00	15:01:00	Ewallet	585.66
109-86-4363	B	Mandalay	Member	Female	Sports and travel	60.08	7	21.0280	441.5880	2019-02-14 00:00:00	11:36:00	Credit card	420.56

3. Add the *time_of_day* column

```

1  SELECT
2      time,
3      (CASE
4          WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
5          WHEN `time` BETWEEN "12:00:00" AND "16:00:00" THEN "Afternoon"
6          ELSE "Evening"
7      END) AS time_of_day
8  FROM sales;

```

time	time_of_day
19:44:00	Evening
12:36:00	Afternoon
17:52:00	Evening
18:02:00	Evening
12:22:00	Afternoon
15:10:00	Afternoon
11:26:00	Morning
15:01:00	Afternoon
11:36:00	Morning

4. Adding *time_of_day* column to our sales table

```
ALTER TABLE sales ADD COLUMN time_of_day VARCHAR(20);
```

```
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
);
```

id	gender	product_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day
1	Male	Food and beverages	45.79	7	16.0265	336.5565	2019-03-13 00:00:00	19:44:00	Credit card	320.53	4.761904716	16.0265	7.0	Evening
2	Female	Health and beauty	62.82	2	6.2820	131.9220	2019-01-17 00:00:00	12:36:00	Ewallet	125.64	4.761904716	6.2820	4.9	Afternoon
3	Male	Sports and travel	25.25	5	6.3125	132.5625	2019-03-20 00:00:00	17:52:00	Cash	126.25	4.761904716	6.3125	6.1	Evening
4	Male	Health and beauty	65.31	7	22.8585	480.0285	2019-03-05 00:00:00	18:02:00	Credit card	457.17	4.761904716	22.8585	4.2	Evening
5	Male	Fashion accessories	21.48	2	2.1480	45.1080	2019-02-27 00:00:00	12:22:00	Ewallet	42.96	4.761904716	2.1480	6.6	Afternoon
6	Male	Sports and travel	69.52	7	24.3320	510.9720	2019-02-01 00:00:00	15:10:00	Credit card	486.64	4.761904716	24.3320	8.5	Afternoon
7	Male	Home and lifestyle	44.34	2	4.4340	93.1140	2019-03-27 00:00:00	11:26:00	Cash	88.68	4.761904716	4.4340	5.8	Morning

5. Similarly adding *day_name* and *month_name* column to our sales table

```
-- Add day_name column
```

```
SELECT
```

```
date,
```

```
DAYNAME(date)
```

```
FROM sales;
```

```
ALTER TABLE sales ADD COLUMN day_name VARCHAR(10);
```

```
UPDATE sales
```

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

```
-- 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);
```

Now we successfully added the three columns to our sales table.

id	gender	product_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day	day_name	month_name
1	Male	Food and beverages	45.79	7	16.0265	336.5565	2019-03-13 00:00:00	19:44:00	Credit card	320.53	4.761904716	16.0265	7.0	Evening	Wednesday	March
2	Female	Health and beauty	62.82	2	6.2820	131.9220	2019-01-17 00:00:00	12:36:00	Ewallet	125.64	4.761904716	6.2820	4.9	Afternoon	Thursday	January
3	Male	Sports and travel	25.25	5	6.3125	132.5625	2019-03-20 00:00:00	17:52:00	Cash	126.25	4.761904716	6.3125	6.1	Evening	Wednesday	March
4	Male	Health and beauty	65.31	7	22.8585	480.0285	2019-03-05 00:00:00	18:02:00	Credit card	457.17	4.761904716	22.8585	4.2	Evening	Tuesday	March
5	Male	Fashion accessories	21.48	2	2.1480	45.1080	2019-02-27 00:00:00	12:22:00	Ewallet	42.96	4.761904716	2.1480	6.6	Afternoon	Wednesday	February
6	Male	Sports and travel	69.52	7	24.3320	510.9720	2019-02-01 00:00:00	15:10:00	Credit card	486.64	4.761904716	24.3320	8.5	Afternoon	Friday	February
7	Male	Home and lifestyle	44.34	2	4.4340	93.1140	2019-03-27 00:00:00	11:26:00	Cash	88.68	4.761904716	4.4340	5.8	Morning	Wednesday	March

Exploratory Data Analysis

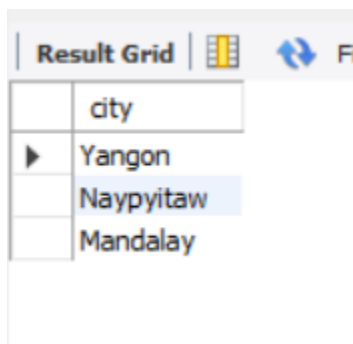
Exploratory data analysis is done to answer the below listed questions and aims of this project.

Generic Question

1. How many unique cities does the data have?

```
-- How many unique cities does the data have?
```

```
SELECT  
    DISTINCT city  
FROM sales;
```



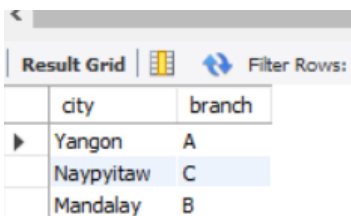
The screenshot shows a 'Result Grid' with a table containing three rows of unique city names. The first row is the header 'city'. The subsequent rows are 'Yangon', 'Naypyitaw', and 'Mandalay'. The 'Naypyitaw' row is highlighted with a blue background.

city
Yangon
Naypyitaw
Mandalay

2. How many unique cities does the data have?

```
-- In which city is each branch?
```

```
SELECT  
    DISTINCT city,  
    branch  
FROM sales;_
```



The screenshot shows a 'Result Grid' with a table containing three rows of unique city and branch combinations. The first row is the header with columns 'city' and 'branch'. The subsequent rows are 'Yangon A', 'Naypyitaw C', and 'Mandalay B'. The 'Naypyitaw C' row is highlighted with a blue background.

city	branch
Yangon	A
Naypyitaw	C
Mandalay	B

Product

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

```
1  -- How many unique product lines does the data have?
2  • SELECT
3      DISTINCT product_line
4  FROM sales;
5
```

Result Grid		Filter Rows:
	product_line	
▶	Food and beverages	
	Health and beauty	
	Sports and travel	
	Fashion accessories	
	Home and lifestyle	
	Electronic accessories	

2. What is the most selling product line?

```
--
-- What is the most selling product line
• SELECT
    SUM(quantity) as qty,
    product_line
FROM sales
GROUP BY product_line
ORDER BY qty DESC;
```

Result Grid		Filter Rows:
	qty	product_line
▶	961	Electronic accessories
	952	Food and beverages
	911	Home and lifestyle
	902	Sports and travel
	902	Fashion accessories
	844	Health and beauty

3. What is the total revenue per month?

```
1  -- What is the total revenue by month
2  SELECT
3      month_name AS month,
4      SUM(total) AS total_revenue
5  FROM sales
6  GROUP BY month_name
7  ORDER BY total_revenue;
8
```

Result Grid		Filter Rows:
month	total_revenue	
February	95727.3765	
March	108867.1500	
January	116291.8680	

4. What month had the largest COGS?

```
1  -- What month had the largest COGS?
2  SELECT
3      month_name AS month,
4      SUM(cogs) AS cogs
5  FROM sales
6  GROUP BY month_name
7  ORDER BY cogs;
```

Result Grid		Filter Rows:
month	cogs	
February	91168.93	
March	103683.00	
January	110754.16	

5. What product line had the largest revenue?

```
1  -- What product line had the largest revenue?
2  SELECT
3      product_line,
4      SUM(total) as total_revenue
5  FROM sales
6  GROUP BY product_line
7  ORDER BY total_revenue DESC;
```

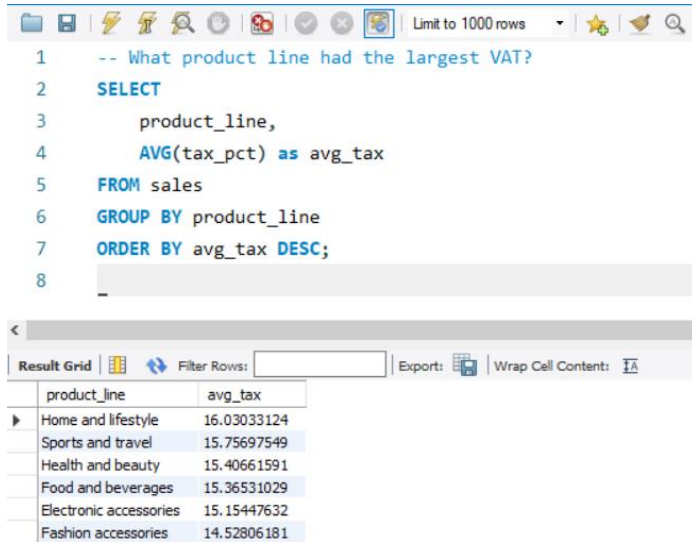
product_line	total_revenue
Food and beverages	56144.8440
Fashion accessories	54305.8950
Sports and travel	53936.1270
Home and lifestyle	53861.9130
Electronic accessories	53783.2365
Health and beauty	48854.3790

6. What is the city with the largest revenue?

```
1  -- What is the city with the largest revenue?
2  SELECT
3      branch,
4      city,
5      SUM(total) AS total_revenue
6  FROM sales
7  GROUP BY city, branch
8  ORDER BY total_revenue;
```

branch	city	total_revenue
B	Mandalay	104534.6085
A	Yangon	105861.0105
C	Naypyitaw	110490.7755

7. What product line had the largest VAT?

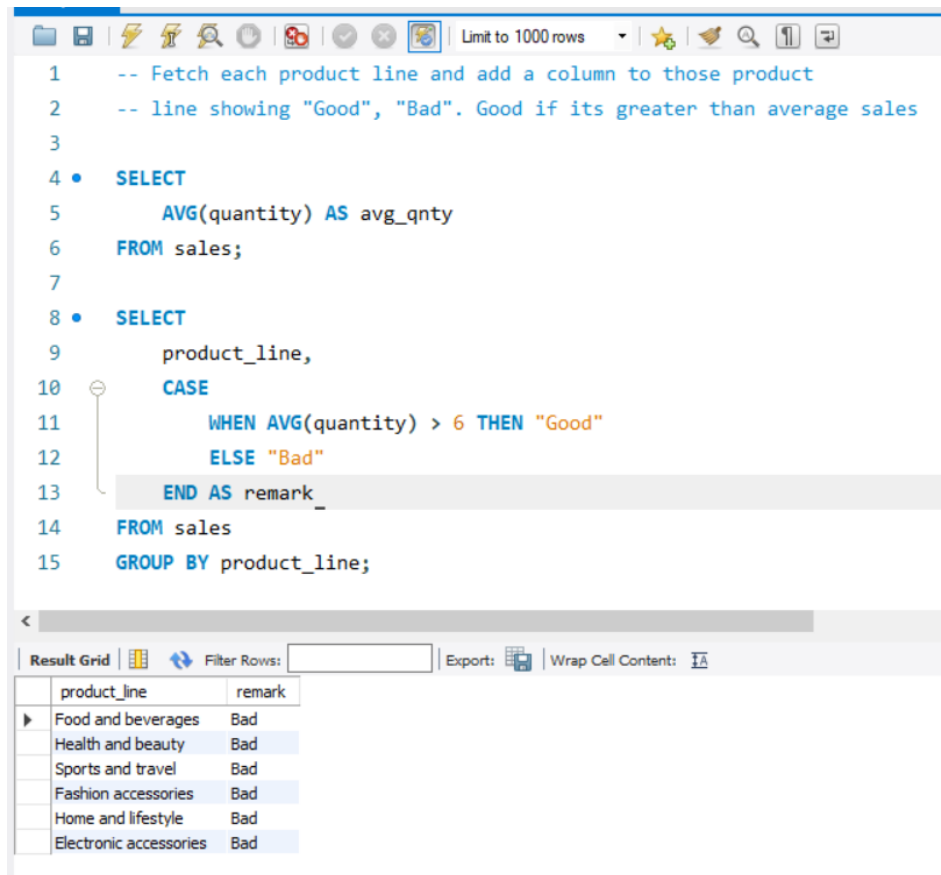


```
1  -- What product line had the largest VAT?
2  SELECT
3      product_line,
4      AVG(tax_pct) as avg_tax
5  FROM sales
6  GROUP BY product_line
7  ORDER BY avg_tax DESC;
```

Result Grid

product_line	avg_tax
Home and lifestyle	16.03033124
Sports and travel	15.75697549
Health and beauty	15.40661591
Food and beverages	15.36531029
Electronic accessories	15.15447632
Fashion accessories	14.52806181

8. Fetch each product line and add a column to those product lines showing "Good", "Bad".



```
1  -- Fetch each product line and add a column to those product
2  -- line showing "Good", "Bad". Good if its greater than average sales
3
4  • SELECT
5      AVG(quantity) AS avg_qty
6  FROM sales;
7
8  • SELECT
9      product_line,
10     CASE
11         WHEN AVG(quantity) > 6 THEN "Good"
12         ELSE "Bad"
13     END AS remark_
14  FROM sales
15  GROUP BY product_line;
```

Result Grid

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

9. Which branch sold more products than average product sold?

```

1  -- Which branch sold more products than average product sold?
2  •  SELECT
3      branch,
4      SUM(quantity) AS qnty
5  FROM sales
6  GROUP BY branch
7  HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);
8
9

```

Result Grid

	branch	qnty
▶	A	1849
	C	1828
	B	1795

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

```

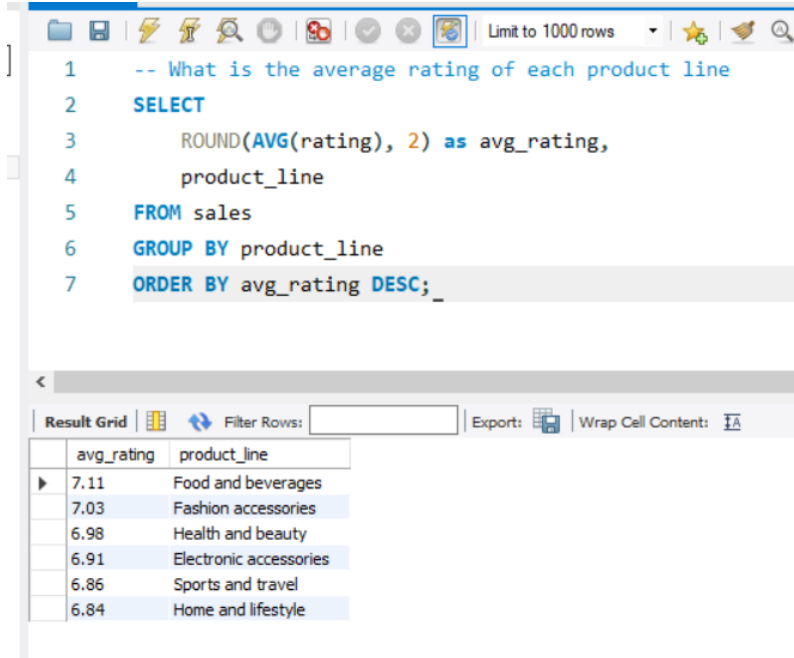
2  SELECT
3      gender,
4      product_line,
5      COUNT(gender) AS total_cnt
6  FROM sales
7  GROUP BY gender, product_line
8  ORDER BY total_cnt DESC;
9

```

Result Grid

	gender	product_line	total_cnt
▶	Female	Fashion accessories	96
	Female	Food and beverages	90
	Male	Health and beauty	88
	Female	Sports and travel	86
	Male	Electronic accessories	86
	Male	Food and beverages	84
	Female	Electronic accessories	83
	Male	Fashion accessories	82
	Male	Home and lifestyle	81
	Female	Home and lifestyle	79
	Male	Sports and travel	77
	Female	Health and beauty	63

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



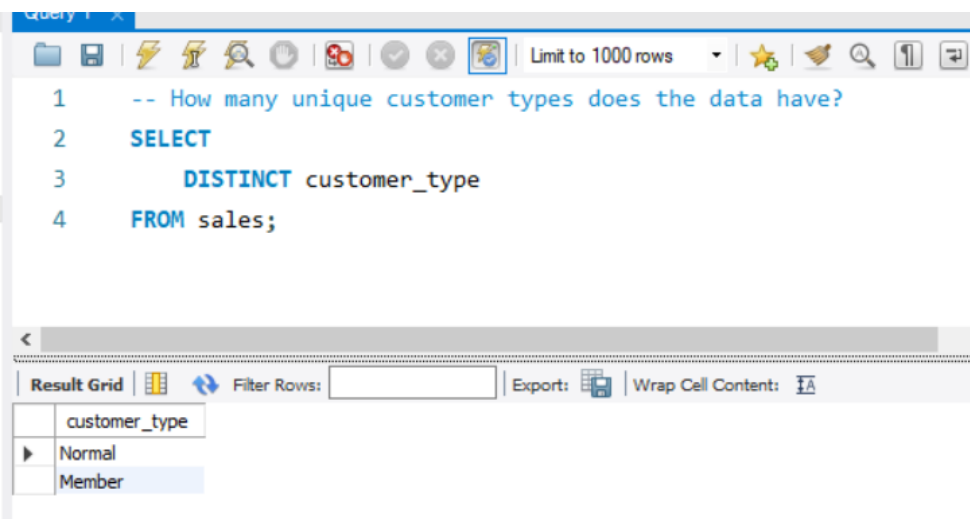
```
1  -- What is the average rating of each product line
2  SELECT
3      ROUND(AVG(rating), 2) as avg_rating,
4      product_line
5  FROM sales
6  GROUP BY product_line
7  ORDER BY avg_rating DESC;
```

Result Grid

	avg_rating	product_line
▶	7.11	Food and beverages
	7.03	Fashion accessories
	6.98	Health and beauty
	6.91	Electronic accessories
	6.86	Sports and travel
	6.84	Home and lifestyle

Customer

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



```
1  -- How many unique customer types does the data have?
2  SELECT
3      DISTINCT customer_type
4  FROM sales;
```

Result Grid

	customer_type
▶	Normal
	Member

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

The screenshot shows a SQL query editor with the following code:

```
1 -- How many unique payment methods does the data have?
2 SELECT
3     DISTINCT payment
4 FROM sales;
```

Below the editor, the results are displayed in a table:

payment
Credit card
Ewallet
Cash

3. What is the most common customer type?

The screenshot shows a SQL query editor with the following code:

```
1 -- What is the most common customer type?
2 SELECT
3     customer_type,
4     count(*) as count
5 FROM sales
6 GROUP BY customer_type
```

Below the editor, the results are displayed in a table:

customer_type	count
Member	499
Normal	496

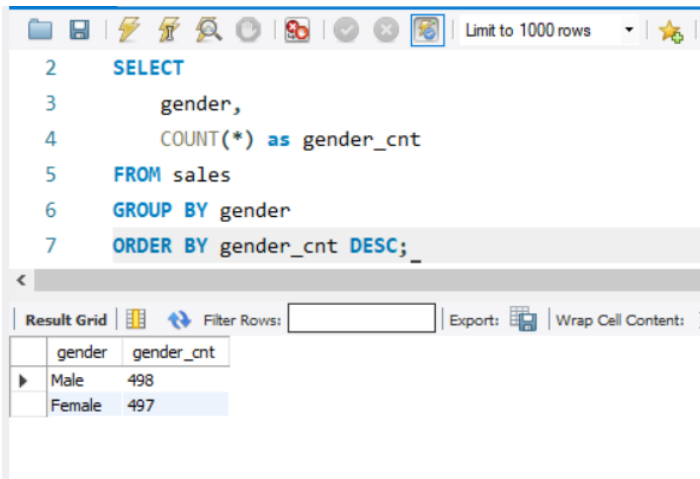
4. Which customer type buys the most?

The screenshot shows a SQL query editor with the following code:

```
1 -- Which customer type buys the most?
2 SELECT
3     customer_type,
4     COUNT(*)
5 FROM sales
6 GROUP BY customer_type;
```

Below the editor, the results are displayed in a table:

customer_type	COUNT(*)
Normal	496
Member	499

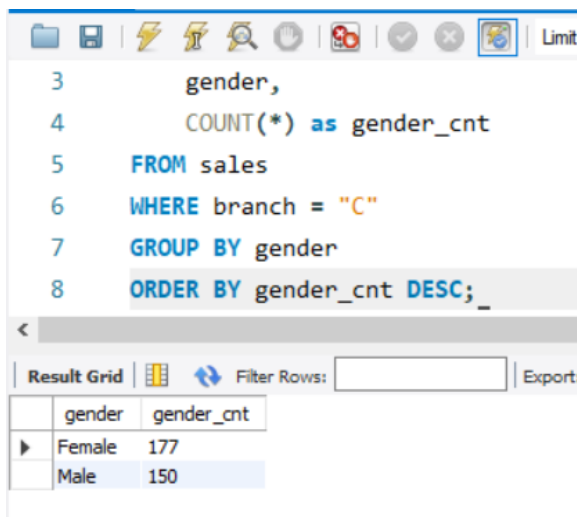
5. What is the gender of most of the customers?

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
2 SELECT
3     gender,
4     COUNT(*) as gender_cnt
5 FROM sales
6 GROUP BY gender
7 ORDER BY gender_cnt DESC;
```

Below the query editor, there is a 'Result Grid' section with a 'Filter Rows' input field and an 'Export' button. The result grid displays the following data:

gender	gender_cnt
Male	498
Female	497

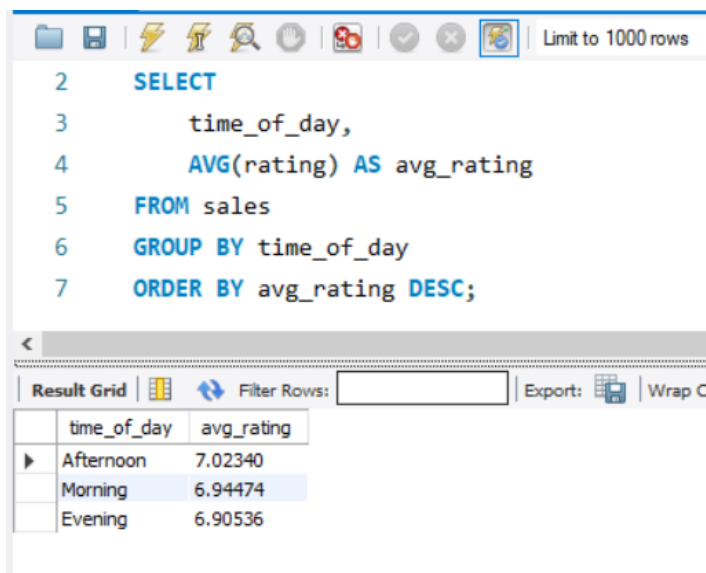
6. What is the gender distribution per branch?

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
3     gender,
4     COUNT(*) as gender_cnt
5 FROM sales
6 WHERE branch = "C"
7 GROUP BY gender
8 ORDER BY gender_cnt DESC;
```

Below the query editor, there is a 'Result Grid' section with a 'Filter Rows' input field and an 'Export' button. The result grid displays the following data:

gender	gender_cnt
Female	177
Male	150

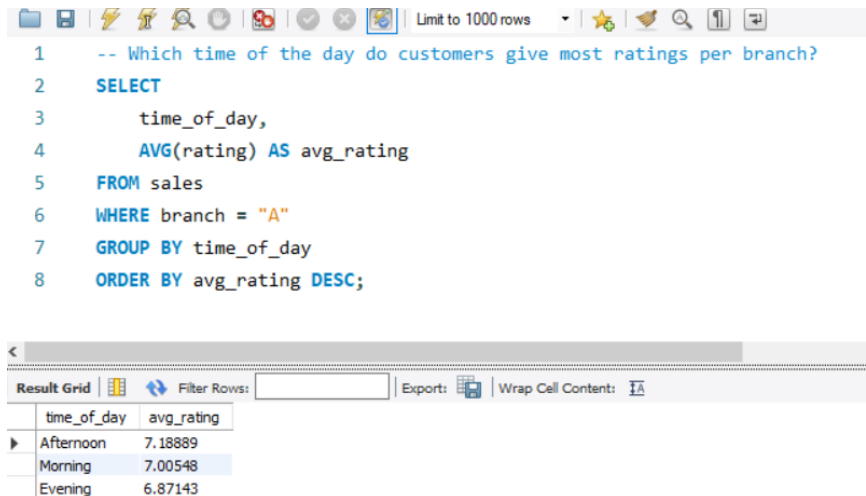
7. Which time of the day do customers give most ratings?

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
2 SELECT
3     time_of_day,
4     AVG(rating) AS avg_rating
5 FROM sales
6 GROUP BY time_of_day
7 ORDER BY avg_rating DESC;
```

Below the query editor, there is a 'Result Grid' section with a 'Filter Rows' input field and an 'Export' button. The result grid displays the following data:

time_of_day	avg_rating
Afternoon	7.02340
Morning	6.94474
Evening	6.90536

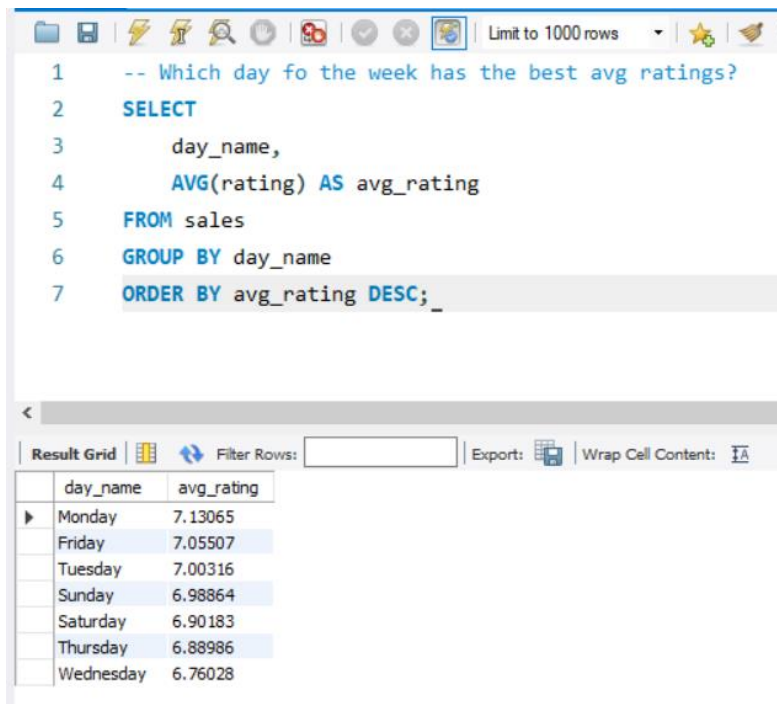
8. Which time of the day do customers give most ratings per branch?

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  -- Which time of the day do customers give most ratings per branch?
2  SELECT
3      time_of_day,
4      AVG(rating) AS avg_rating
5  FROM sales
6  WHERE branch = "A"
7  GROUP BY time_of_day
8  ORDER BY avg_rating DESC;
```

Below the query editor, the results are displayed in a table with the following data:

time_of_day	avg_rating
Afternoon	7.18889
Morning	7.00548
Evening	6.87143

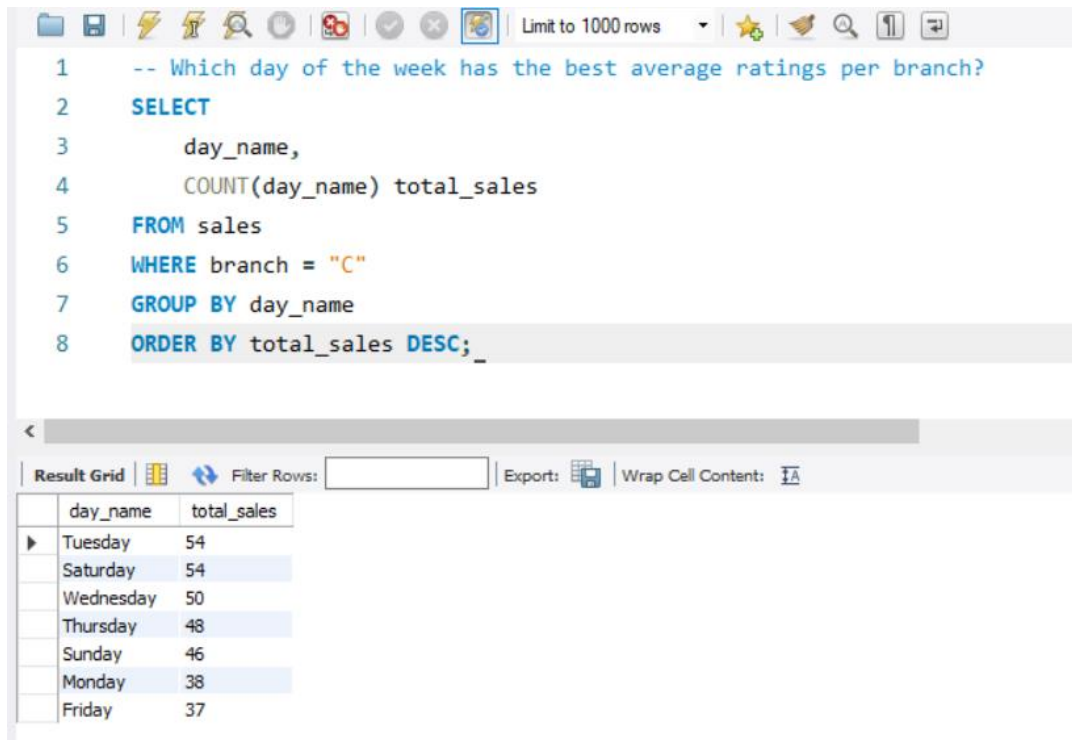
9. Which day of the week has the best avg ratings?

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  -- Which day of the week has the best avg ratings?
2  SELECT
3      day_name,
4      AVG(rating) AS avg_rating
5  FROM sales
6  GROUP BY day_name
7  ORDER BY avg_rating DESC;
```

Below the query editor, the results are displayed in a table with the following data:

day_name	avg_rating
Monday	7.13065
Friday	7.05507
Tuesday	7.00316
Sunday	6.98864
Saturday	6.90183
Thursday	6.88986
Wednesday	6.76028

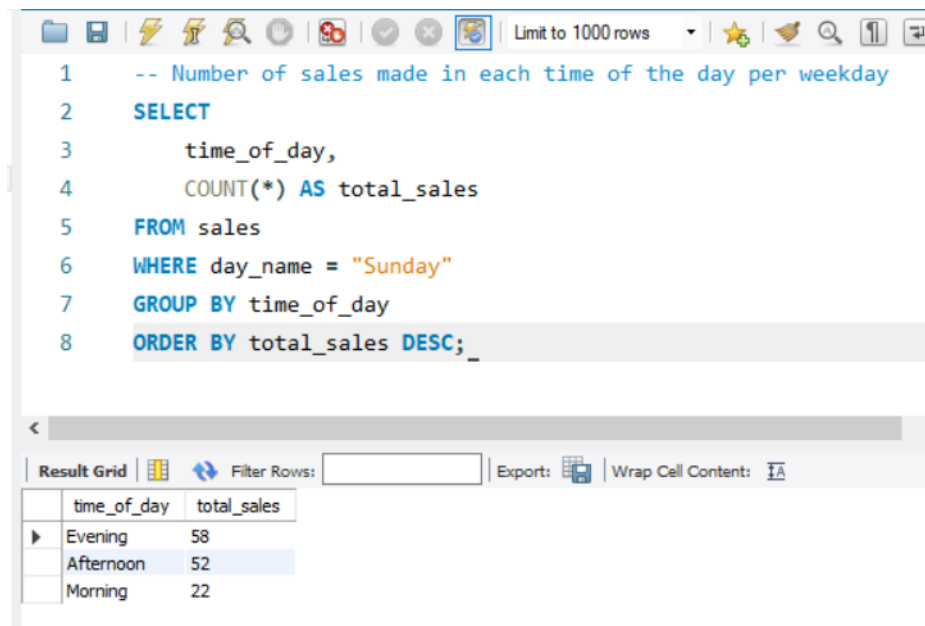
10. Which day of the week has the best average ratings per branch?

```
1  -- Which day of the week has the best average ratings per branch?
2  SELECT
3      day_name,
4      COUNT(day_name) total_sales
5  FROM sales
6  WHERE branch = "C"
7  GROUP BY day_name
8  ORDER BY total_sales DESC;
```

Result Grid

	day_name	total_sales
▶	Tuesday	54
	Saturday	54
	Wednesday	50
	Thursday	48
	Sunday	46
	Monday	38
	Friday	37

Sales

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

```
1  -- Number of sales made in each time of the day per weekday
2  SELECT
3      time_of_day,
4      COUNT(*) AS total_sales
5  FROM sales
6  WHERE day_name = "Sunday"
7  GROUP BY time_of_day
8  ORDER BY total_sales DESC;
```

Result Grid

	time_of_day	total_sales
▶	Evening	58
	Afternoon	52
	Morning	22

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

```
1  -- Which of the customer types brings the most revenue?
2  SELECT
3      customer_type,
4      SUM(total) AS total_revenue
5  FROM sales
6  GROUP BY customer_type
7  ORDER BY total_revenue;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	customer_type	total_revenue
▶	Normal	157261.2930
	Member	163625.1015

3. Which city has the largest tax/VAT percent?

```
1  -- Which city has the largest tax/VAT percent?
2  SELECT
3      city,
4      ROUND(AVG(tax_pct), 2) AS avg_tax_pct
5  FROM sales
6  GROUP BY city
7  ORDER BY avg_tax_pct DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	city	avg_tax_pct
▶	Naypyitaw	16.09
	Mandalay	15.13
	Yangon	14.87

4. Which customer type pays the most in VAT?

```
1  -- Which customer type pays the most in VAT?
2  SELECT
3      customer_type,
4      AVG(tax_pct) AS total_tax
5  FROM sales
6  GROUP BY customer_type
7  ORDER BY total_tax;
8  --
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	customer_type	total_tax
▶	Normal	15.09805040
	Member	15.61457214