

In this project, I analyzed Walmart's sales data using SQL to extract key business insights. I created structured tables from raw CSV data and performed aggregation, grouping, and filtering operations. The analysis focused on total revenue, branch-wise and city-wise performance, customer types, payment method trends, product line popularity, and customer satisfaction ratings. Through SQL queries, I identified top-performing branches, best-selling products, peak sales periods, and the relationship between customer type and total spending. This project strengthened my skills in data cleaning, SQL joins, subqueries, time-based analysis, and performance optimization.

how many unique cities does the data have?

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

Result Grid		Filter Rows:	
	city		
▶	Yangon		
	Naypyitaw		
	Mandalay		

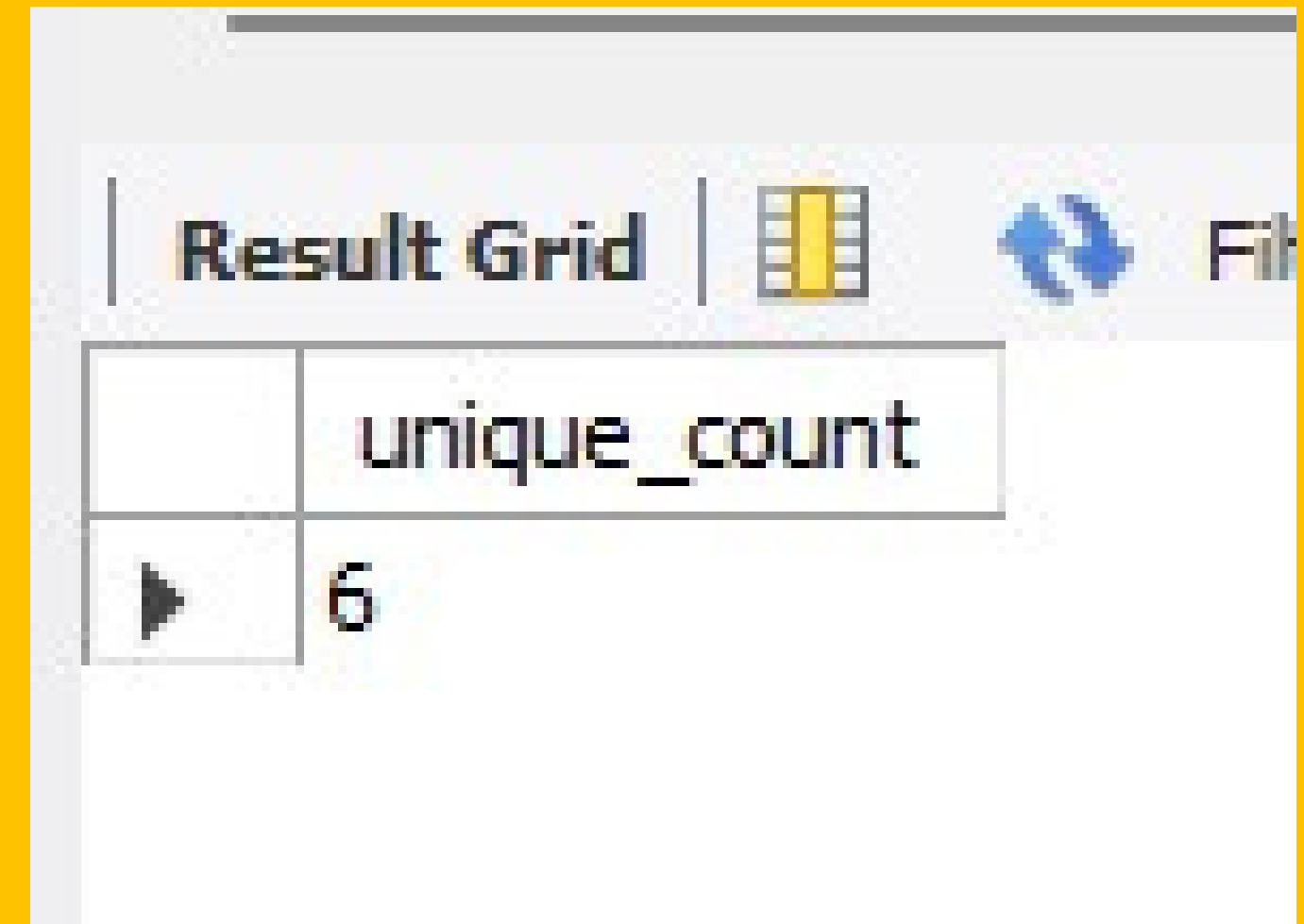
in which city each branch?

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

Result Grid			Filter Rows:
	city	branch	
▶	Yangon	A	
	Naypyitaw	C	
	Mandalay	B	

how many unique product lines does the data
have?

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





A screenshot of a database query result grid. The grid has a header row with the column name 'unique_count' and a data row with the value '6'. The grid is titled 'Result Grid' and has a 'Filter' button on the right.

	unique_count
▶	6

which day of the week has the best avg rating per branch?

```
SELECT  
day_name, AVG(rating) AS  
avg_rating  
FROM  
sales  
WHERE  
branch = 'a'  
GROUP BY day_name  
ORDER BY avg_rating DESC;
```

Result Grid				 Filter Rows:
	day_name	avg_rating		
▶	Friday	7.31200		
	Monday	7.09792		
	Sunday	7.07885		
	Tuesday	7.05882		
	Thursday	6.95870		
	Wednesday	6.84286		
	Saturday	6.74600		

what is the most selling product line?

```
SELECT  
  product_line,  
  COUNT(product_line) AS COUNT  
FROM  
  sales  
  
GROUP BY product_line  
ORDER BY COUNT DESC;
```

Result Grid			Filter Rows:
	product_line	COUNT	
▶	Fashion accessories	178	
	Food and beverages	174	
	Electronic accessories	169	
	Sports and travel	163	
	Home and lifestyle	160	
	Health and beauty	151	

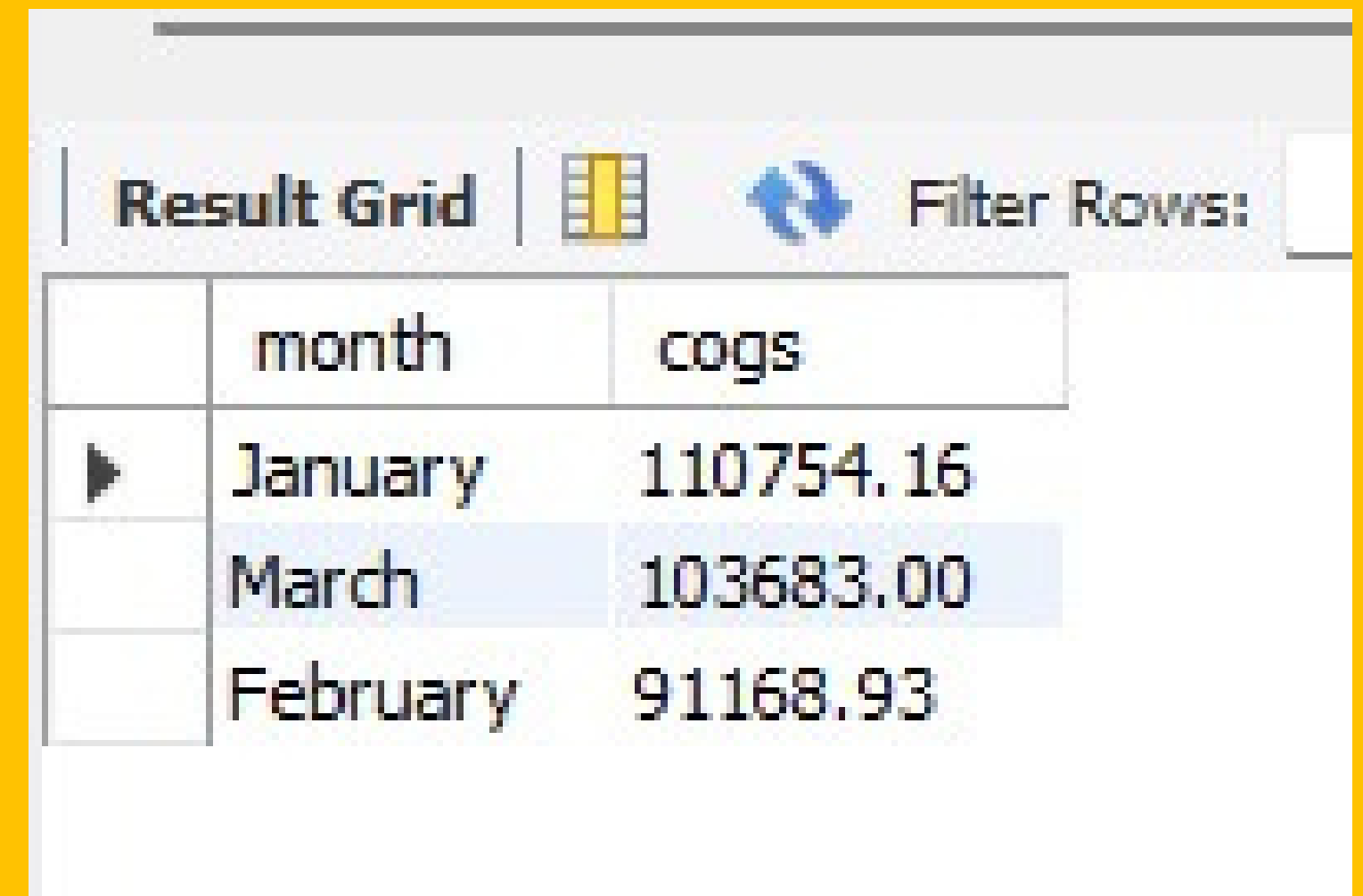
what is the total revenue by month?

```
SELECT
month_name AS month, SUM(total) AS
total_revenue
FROM
sales
GROUP BY month_name
ORDER BY total_revenue DESC;
```

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

what month had the leargest cogs?

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



A screenshot of a database application's 'Result Grid' window. The window has a title bar and a toolbar with icons for 'Result Grid', a grid icon, a refresh icon, and a 'Filter Rows:' dropdown. The grid contains three columns: an empty column, 'month', and 'cogs'. The data is sorted by 'cogs' in descending order. The first row is 'January' with a value of 110754.16. The second row is 'March' with a value of 103683.00. The third row is 'February' with a value of 91168.93.

	month	cogs
▶	January	110754.16
	March	103683.00
	February	91168.93

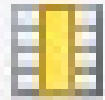

what product line had largest revenue?

```
SELECT
product_line, SUM(total) AS
total_revenue
FROM
sales
GROUP BY product_line
ORDER BY total_revenue DESC;
```

Result Grid			Filter Rows:
	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	

what is the city with largest revenue?

```
SELECT
city, branch, SUM(total) AS total_revenue
FROM
sales
GROUP BY city , branch
ORDER BY total_revenue DESC;
```

Result Grid   Filter Rows: <input data-bbox="3045 979 3262 1095" type="text"/>			
	city	branch	total_revenue
▶	Naypyitaw	C	110490.7755
	Yangon	A	105861.0105
	Mandalay	B	104534.6085

what product line had avg vat(value add tax)?

```
SELECT
product_line, AVG(vat) AS avg_tax
FROM
sales
GROUP BY product_line
ORDER BY avg_tax DESC;
```

Result Grid			Filter Rows:
	product_line	avg_tax	
▶	Home and lifestyle	16.0313	
	Sports and travel	15.7301	
	Health and beauty	15.4106	
	Food and beverages	15.3621	
	Electronic accessories	15.1716	
	Fashion accessories	14.5506	



which day of the week has the best avg rating per branch?

```
SELECT
day_name, AVG(rating) AS avg_rating
FROM
sales
WHERE
branch = 'a'
GROUP BY day_name
ORDER BY avg_rating DESC;
```

Result Grid			Filter Rows:
	day_name	avg_rating	
►	Friday	7.31200	
	Monday	7.09792	
	Sunday	7.07885	
	Tuesday	7.05882	
	Thursday	6.95870	
	Wednesday	6.84286	
	Saturday	6.74600	



what is the most common product line by gender?

```
SELECT
  gender, product_line,
  COUNT(gender) AS total_count
FROM
  sales
GROUP BY gender , product_line
ORDER BY total_count DESC;
```

Result Grid   Filter Rows: <input type="text"/>			
	gender	product_line	total_count
▶	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	Home and lifestyle	77

which day of the week has the best avg rating per branch?

```
SELECT
day_name, AVG(rating) AS avg_rating
FROM
sales
WHERE
branch = 'a'
GROUP BY day_name
ORDER BY avg_rating DESC;
```

Result Grid  Filter Rows: 		
	day_name	avg_rating
▶	Friday	7.31200
	Monday	7.09792
	Sunday	7.07885
	Tuesday	7.05882
	Thursday	6.95870
	Wednesday	6.84286
	Saturday	6.74600



number of sales made in each time of day per week?

```
SELECT  
time_of_day, COUNT(*) AS total_sales  
FROM  
sales  
WHERE  
day_name = 'Monday'  
GROUP BY time_of_day  
ORDER BY total_sales;
```

Result Grid			Filter Rows:
	time_of_day	total_sales	
▶	Morning	20	
	Afternoon	48	
	Evening	56	

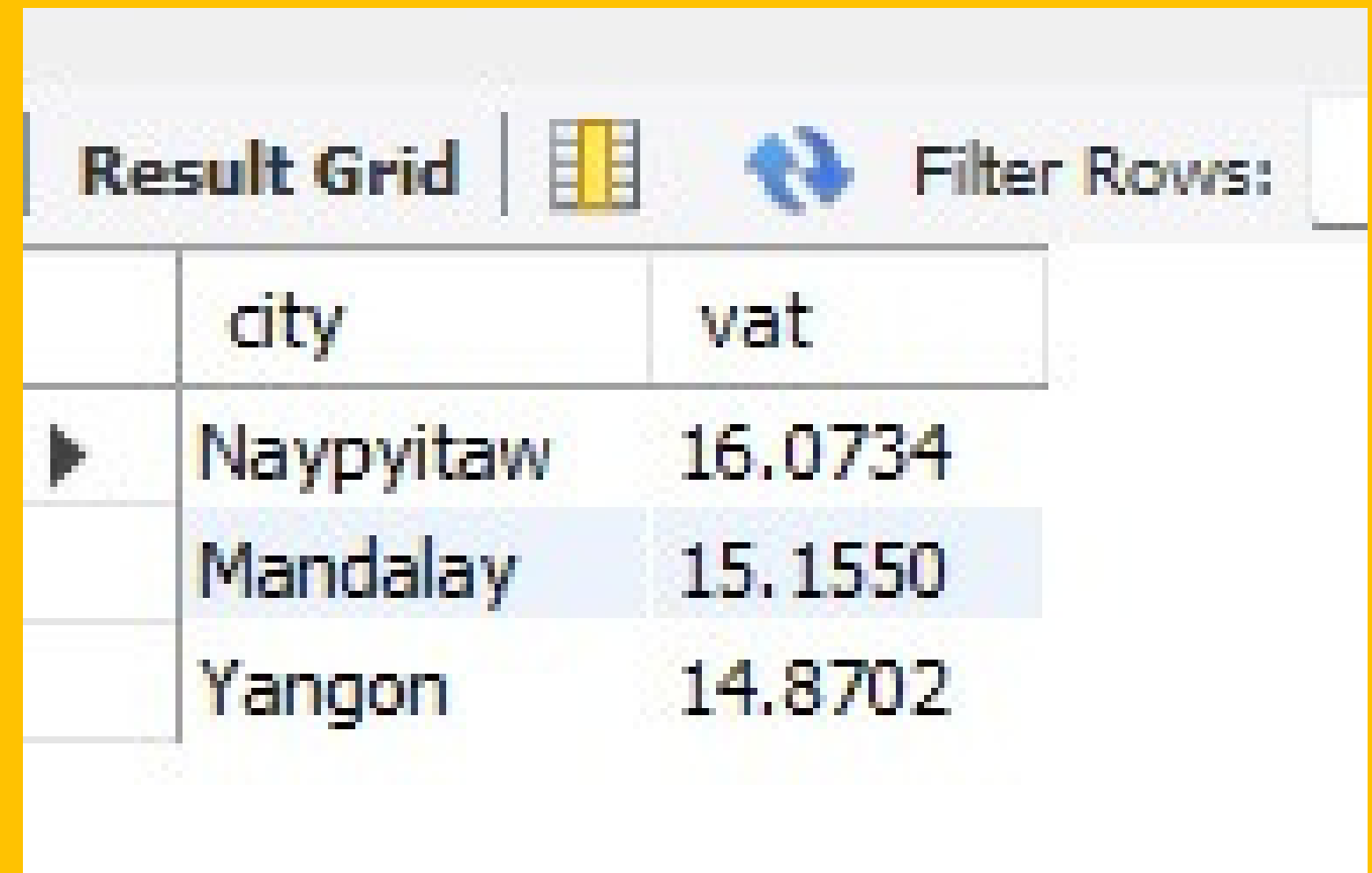
which of the customer types brings the most revenue?

```
SELECT  
customer_type, SUM(total) AS  
total_revenue  
FROM  
sales  
GROUP BY customer_type  
ORDER BY total_revenue DESC;
```

Result Grid   Filter Rows: <input type="text"/>		
	customer_type	total_revenue
*	Member	163625.1015
	Normal	157261.2930

which city has the largest avg tax %/ vat (value added tax)?

```
SELECT
city, AVG(vat) AS vat
FROM
sales
GROUP BY city
ORDER BY vat DESC;
```



The screenshot shows a 'Result Grid' window with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows:' input field. The grid displays the results of the SQL query, ordered by average VAT percentage in descending order. The first three rows are visible: Naypyitaw (16.0734), Mandalay (15.1550), and Yangon (14.8702). The 'Mandalay' row is highlighted with a blue background.

	city	vat
▶	Naypyitaw	16.0734
	Mandalay	15.1550
	Yangon	14.8702

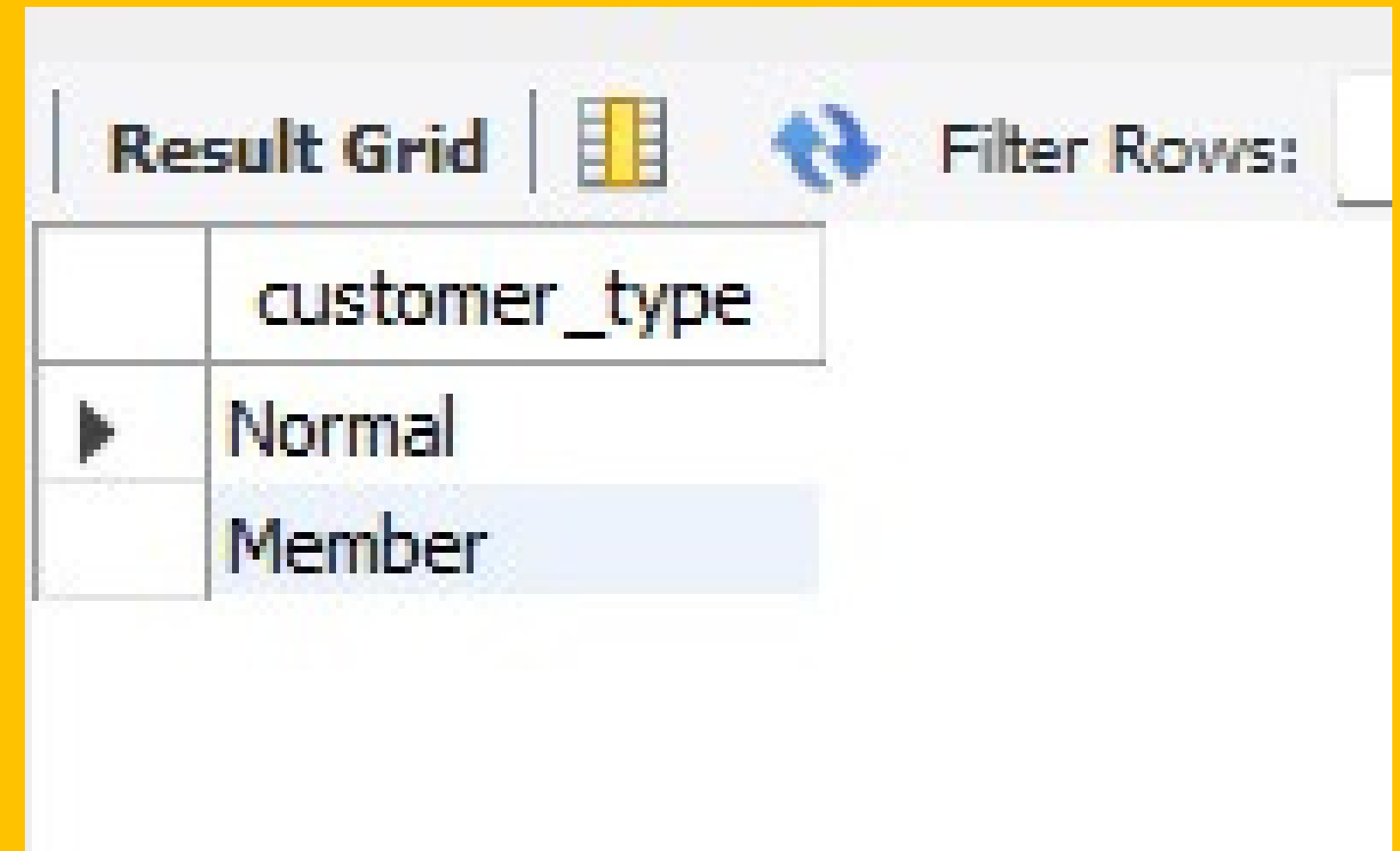
which day of the week has the best avg rating per branch?

```
SELECT
day_name, AVG(rating) AS avg_rating
FROM
sales
WHERE
branch = 'a'
GROUP BY day_name
ORDER BY avg_rating DESC;
```

Result Grid			Filter Rows:
	day_name	avg_rating	
▶	Friday	7.31200	
	Monday	7.09792	
	Sunday	7.07885	
	Tuesday	7.05882	
	Thursday	6.95870	
	Wednesday	6.84286	
	Saturday	6.74600	

how many unique customer type does the data have?

```
SELECT DISTINCT  
customer_type  
FROM  
sales;
```

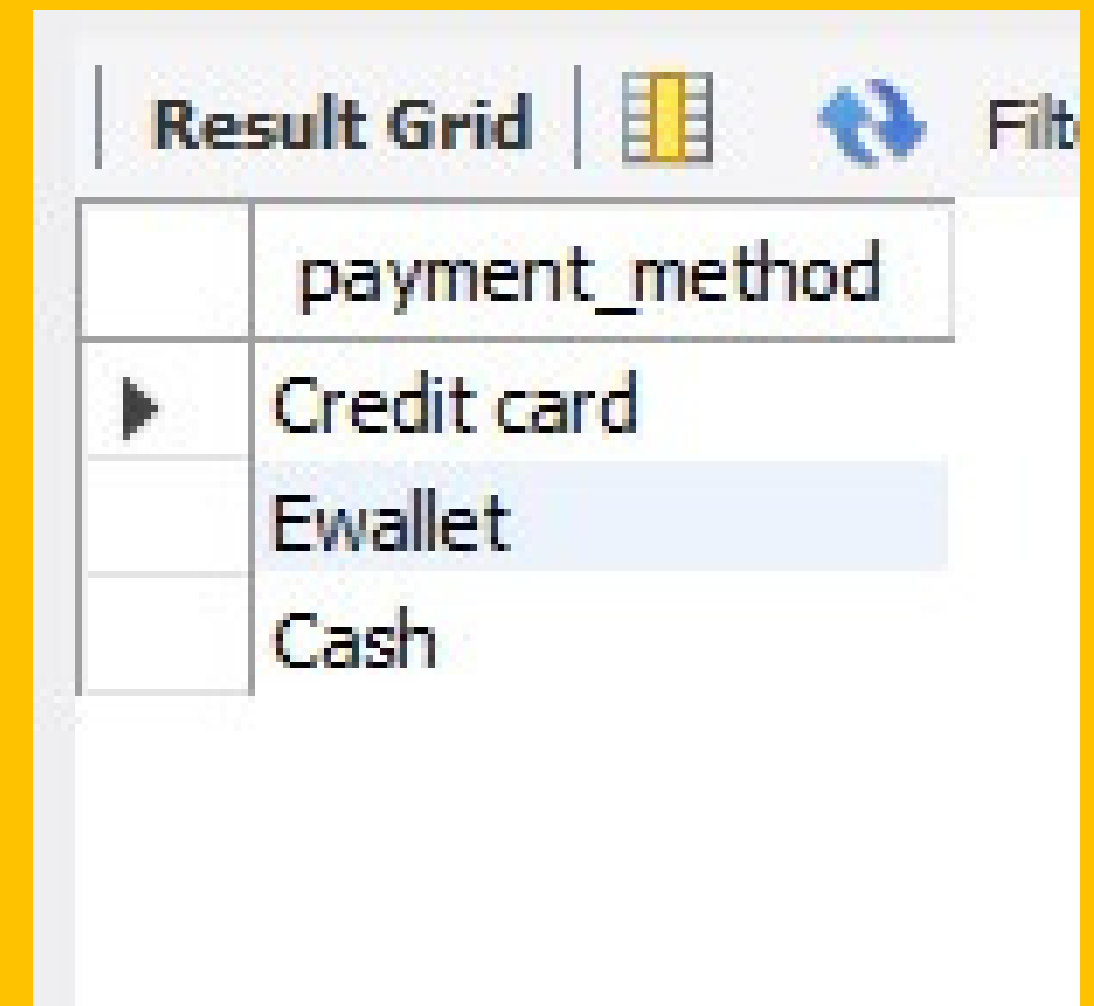


The screenshot shows a database query result grid. The header row is labeled 'customer_type'. There are two data rows: 'Normal' and 'Member'. The 'Member' row is highlighted. The interface includes a 'Result Grid' tab, a grid icon, a 'Filter Rows' button, and a search input field.

	customer_type
▶	Normal
	Member

how many unique payment method the data have?

```
SELECT DISTINCT  
payment_method  
FROM  
sales;
```



The screenshot shows a 'Result Grid' window with a table of unique payment methods. The table has two columns: an expandable column with a right-pointing triangle and a text column. The text column contains three entries: 'Credit card', 'Ewallet', and 'Cash'. The 'Ewallet' row is highlighted with a light blue background. Above the table, there are icons for a grid, a refresh/clear icon, and a 'Filter' label.

	payment_method
▶	Credit card
	Ewallet
	Cash

which customer type buy the most?

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

Result Grid			Filter Rows:
	customer_type	customer_count	
▶	Normal	496	
	Member	499	

what is the gender distribution per branch?

```
SELECT
gender, COUNT(*) AS gender_count
FROM
sales
WHERE
branch = 'a'
GROUP BY gender
ORDER BY gender_count;
```

Result Grid			Filter Rows:
	gender	gender_count	
▶	Female	160	
	Male	179	

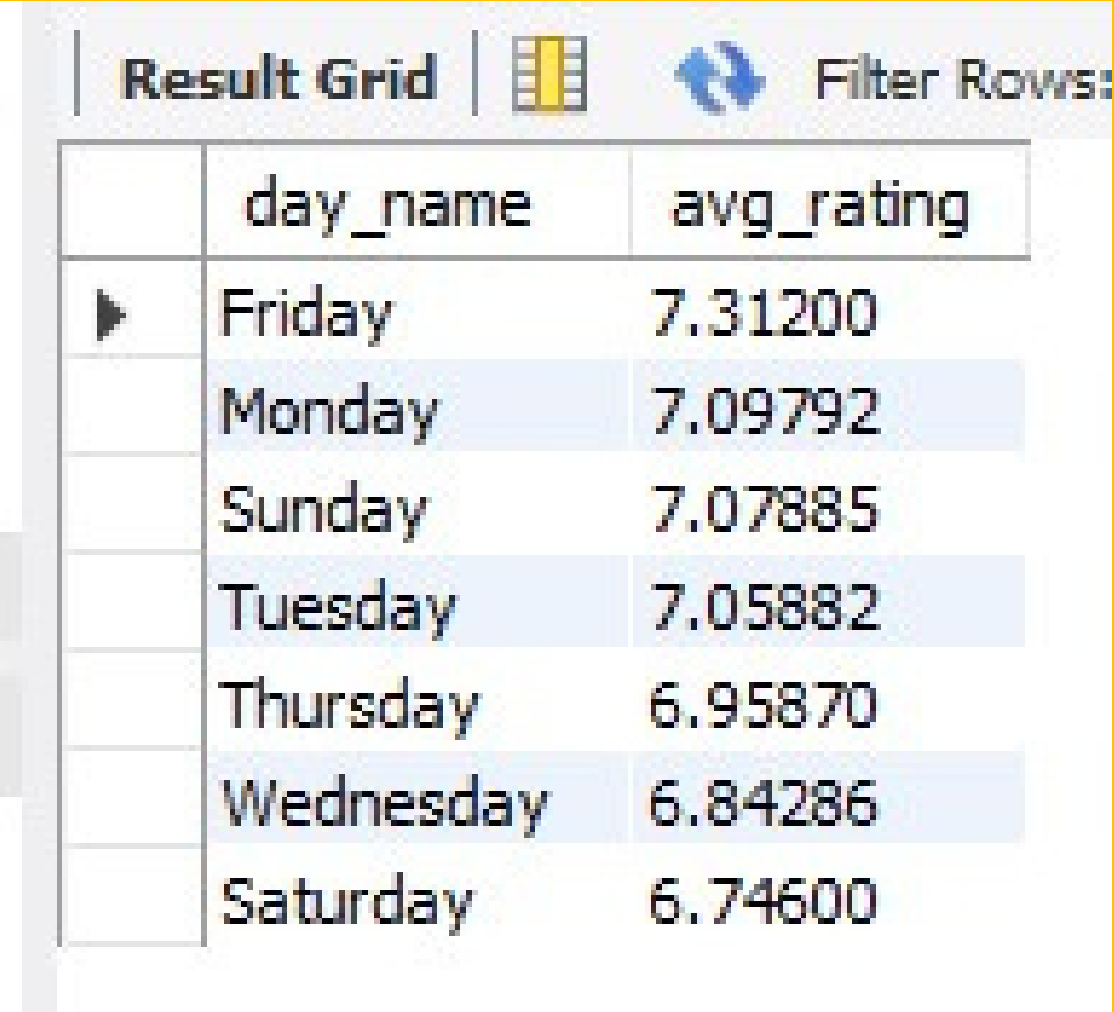
what is the gender type buy the most?

```
SELECT
gender, COUNT(*) AS gender
FROM
sales
GROUP BY gender
ORDER BY gender DESC;
```

Result Grid			Filter Rows
	gender	gender	
▶	Male	498	
	Female	497	

which day of the week has the best avg rating per branch?

```
SELECT
day_name, AVG(rating) AS avg_rating
FROM
sales
WHERE
branch = 'a'
GROUP BY day_name
ORDER BY avg_rating DESC;
```

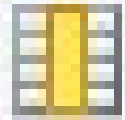



The screenshot shows a 'Result Grid' interface with a table of data. The table has two columns: 'day_name' and 'avg_rating'. The data is sorted in descending order of average rating. The rows are: Friday (7.31200), Monday (7.09792), Sunday (7.07885), Tuesday (7.05882), Thursday (6.95870), Wednesday (6.84286), and Saturday (6.74600). The interface includes a 'Filter Rows' button and a grid icon.

	day_name	avg_rating
▶	Friday	7.31200
	Monday	7.09792
	Sunday	7.07885
	Tuesday	7.05882
	Thursday	6.95870
	Wednesday	6.84286
	Saturday	6.74600

which time of the day customer give most rating par branch?

```
SELECT  
time_of_day, AVG(rating) AS avg_rating  
FROM  
sales  
WHERE  
branch = 'a'  
GROUP BY time_of_day  
ORDER BY avg_rating DESC;
```

Result Grid   Filter Rows		
	time_of_day	avg_rating
▶	Afternoon	7.18889
	Morning	7.00548
	Evening	6.87143

which day of the week has the best avg rating?

```
SELECT
day_name, AVG(rating) AS avg_rating
FROM
sales
GROUP BY day_name
ORDER BY avg_rating DESC;
```

Result Grid			Filter Rows:
	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	