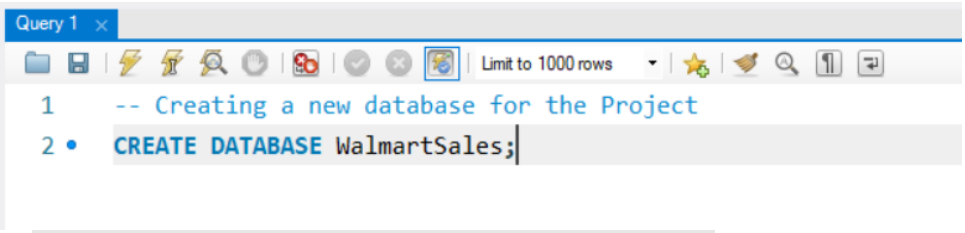
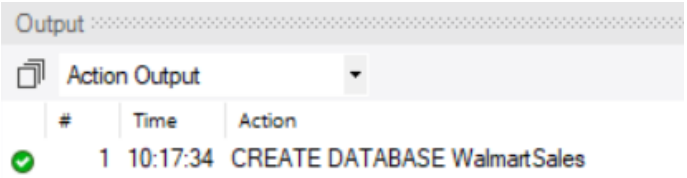
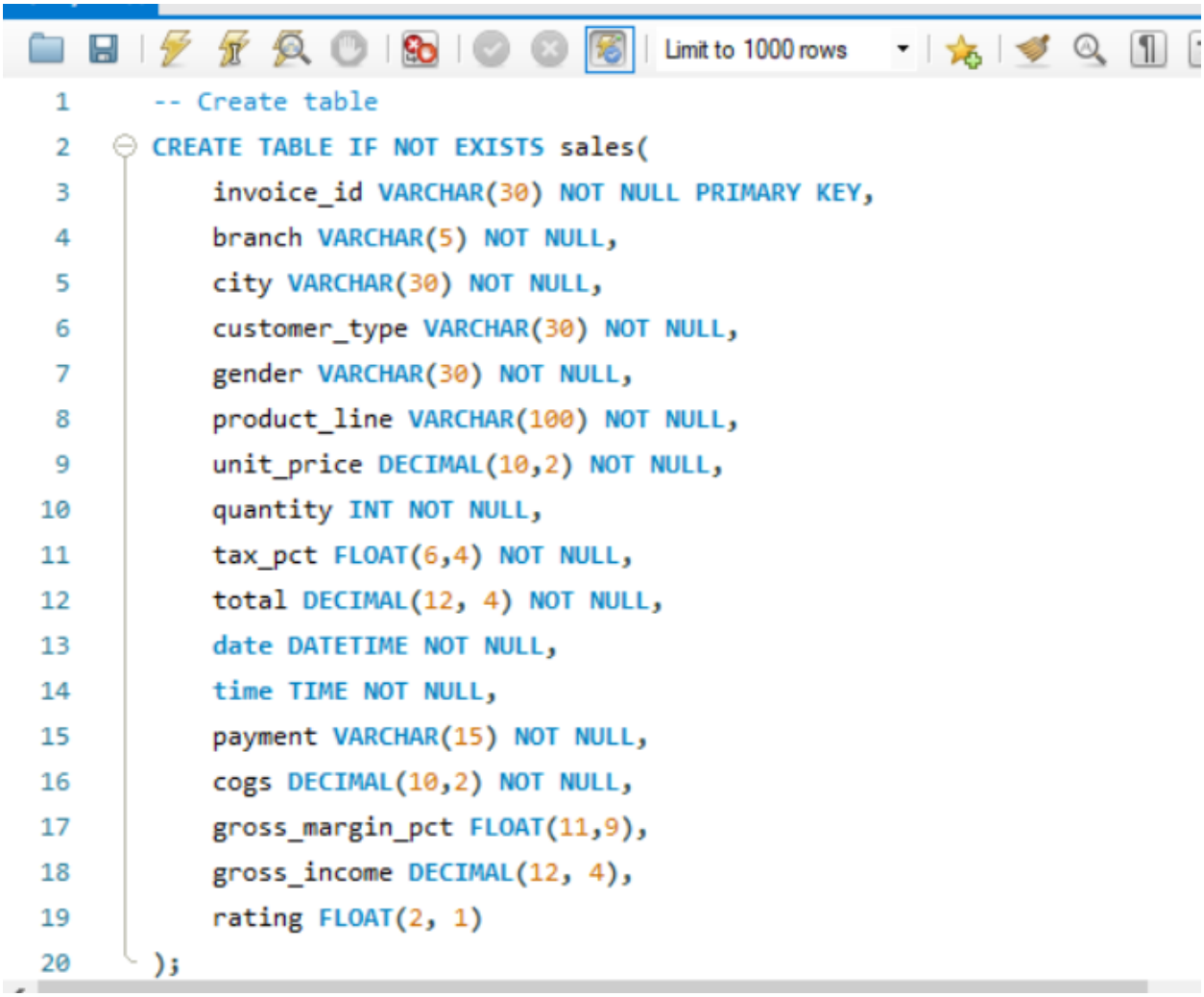
SQL Queries for Walmart Project

1. **Creating a new Database WalmartSales for the Project**





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



```sql
SELECT
    *
FROM sales;
```

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

```sql
SELECT
    time,
    (CASE
        WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
        WHEN `time` BETWEEN "12:00:00"AND "16:00:00" THEN "Afternoon"
        ELSE "Evening"
    END) AS time_of_day
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

4. Adding *time_of_day* column to our sales table

```sql
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
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

gender	product_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day
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
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
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
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
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
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
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

```sql
-- 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);
```

```sql
-- 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.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day	day_name	month_name
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
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
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
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
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
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
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

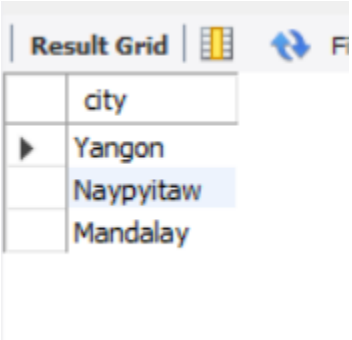
Exploratory Data Analysis

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

Generic Question

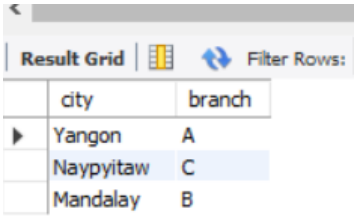
1. **How many unique cities does the data have?**

```
-- How many unique cities does the data have?
SELECT
    DISTINCT city
FROM sales;
```



2. **How many unique cities does the data have?**

```
-- In which city is each branch?
SELECT
    DISTINCT city,
    branch
FROM sales;
```



Product

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

```sql
-- How many unique product lines does the data have?
SELECT
    DISTINCT product_line
FROM sales;
```

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?**

```sql
-- 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
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?

```sql
-- 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;
```

Result Grid | Filter Rows:

month	total_revenue
February	95727.3765
March	108867.1500
January	116291.8680

4. What month had the largest COGS?

```sql
-- What month had the largest COGS?
SELECT
    month_name AS month,
    SUM(cogs) AS cogs
FROM sales
GROUP BY month_name
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?

```sql
-- 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;
```

Result Grid | Filter Rows: | Export:

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?

Limit to 1000 rows

```sql
-- 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;
```

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

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

7. What product line had the largest VAT?

```sql
-- What product line had the largest VAT?
SELECT
    product_line,
    AVG(tax_pct) as avg_tax
FROM sales
GROUP BY product_line
ORDER BY avg_tax DESC;
```

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".

```sql
-- Fetch each product line and add a column to those product
-- line showing "Good", "Bad". Good if its greater than average sales

SELECT
    AVG(quantity) AS avg_qnty
FROM sales;

SELECT
    product_line,
    CASE
        WHEN AVG(quantity) > 6 THEN "Good"
        ELSE "Bad"
    END AS remark
FROM sales
GROUP BY product_line;
```

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?

```sql
-- Which branch sold more products than average product sold?
SELECT
    branch,
    SUM(quantity) AS qnty
FROM sales
GROUP BY branch
HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);
```

branch	qnty
A	1849
C	1828
B	1795

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

```sql
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	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?

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

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?**

```sql
-- How many unique customer types does the data have?
SELECT
    DISTINCT customer_type
FROM sales;
```

customer_type
Normal
Member

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

```
1   -- How many unique payment methods does the data have?
2   SELECT
3       DISTINCT payment
4   FROM sales;
5
```

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

payment
Credit card
Ewallet
Cash

3. What is the most common customer type?

```
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
```

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

customer_type	count
Member	499
Normal	496

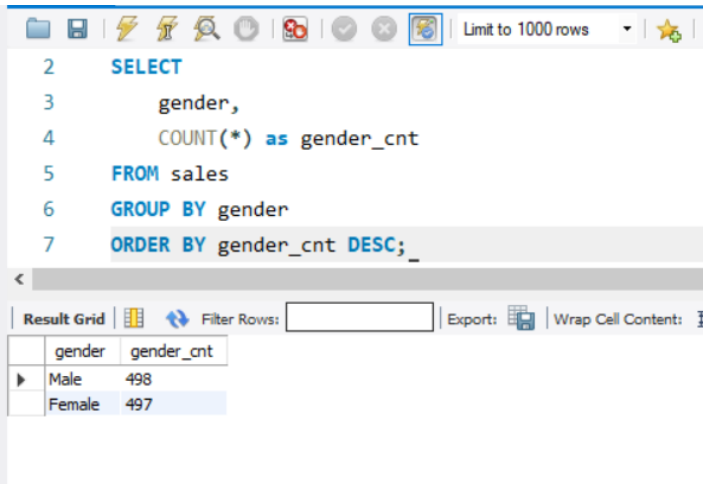
4. Which customer type buys the most?

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

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

customer_type	COUNT(*)
Normal	496
Member	499

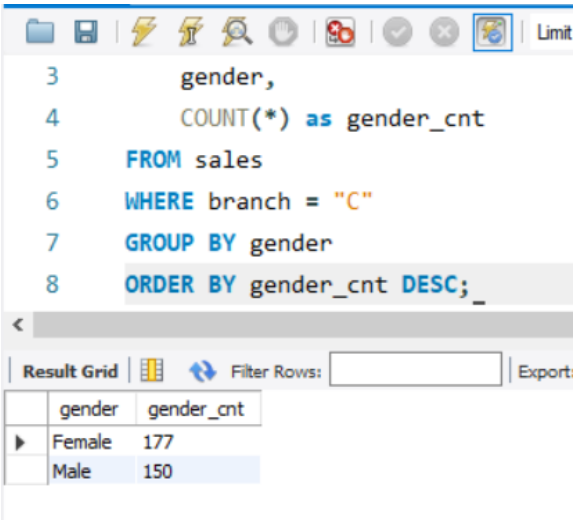
5. What is the gender of most of the customers?



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

gender	gender_cnt
Male	498
Female	497

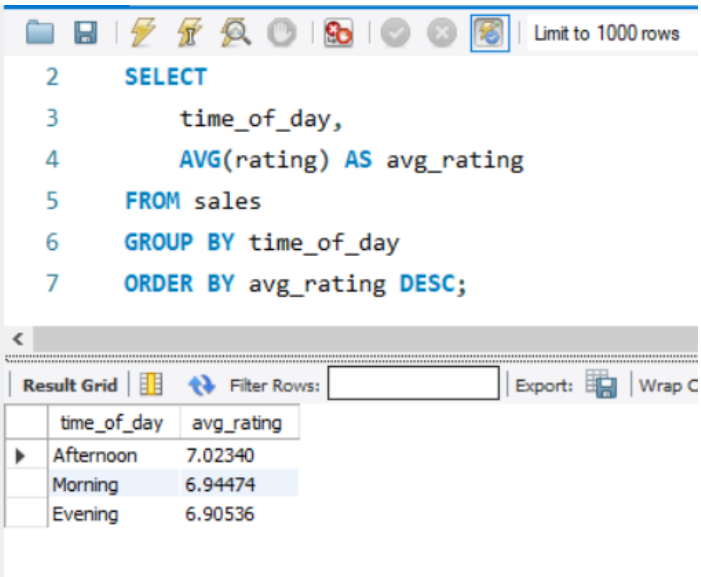
6. What is the gender distribution per branch?



```
    gender,
    COUNT(*) as gender_cnt
FROM sales
WHERE branch = "C"
GROUP BY gender
ORDER BY gender_cnt DESC;
```

gender	gender_cnt
Female	177
Male	150

7. 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	7.02340
Morning	6.94474
Evening	6.90536

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

```sql
-- Which time of the day do customers give most ratings per 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;
```

time_of_day	avg_rating
Afternoon	7.18889
Morning	7.00548
Evening	6.87143

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

```sql
-- Which day fo 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	7.13065
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?

```sql
-- Which day of the week has the best average ratings per branch?
SELECT
    day_name,
    COUNT(day_name) total_sales
FROM sales
WHERE branch = "C"
GROUP BY day_name
ORDER BY total_sales DESC;
```

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

```sql
-- Number of sales made in each time of the day per weekday
SELECT
    time_of_day,
    COUNT(*) AS total_sales
FROM sales
WHERE day_name = "Sunday"
GROUP BY time_of_day
ORDER BY total_sales DESC;
```

time_of_day	total_sales
Evening	58
Afternoon	52
Morning	22

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

```sql
-- 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	157261.2930
Member	163625.1015

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

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

city	avg_tax_pct
Naypyitaw	16.09
Mandalay	15.13
Yangon	14.87

4. Which customer type pays the most in VAT?

```sql
-- Which customer type pays the most in VAT?
SELECT
    customer_type,
    AVG(tax_pct) AS total_tax
FROM sales
GROUP BY customer_type
ORDER BY total_tax;
```

customer_type	total_tax
Normal	15.09805040
Member	15.61457214