Business Problems:

1. Find different payment method and number of transactions, number of quantity sold

```
 SELECT payment_method,
    -> COUNT(*)
    -> , SUM(quantity)
    -> FROM walmart
    -> GROUP BY
    -> payment_method;
```

2. Identity the highest-rated category in each branch, displaying the branch, category

```
 SELECT *
    -> FROM (
    ->     SELECT branch, category, AVG(rating) AS avg_rating,
    ->            RANK() OVER (PARTITION BY branch ORDER BY AVG(rating) DESC) AS rank_
    ->     FROM walmart
    ->     GROUP BY branch, category
    -> ) as ranked
    -> WHERE rank_ = 1;
```

3. Identify the busiest day for each branch based on the number of transactions

```
SELECT branch, day_name, no_transactions, rnk
FROM (
    SELECT
        Branch AS branch,
        DATE_FORMAT(STR_TO_DATE(date, '%d/%m/%y'), '%W') AS day_name,
        COUNT(*) AS no_transactions,
        RANK() OVER (PARTITION BY Branch ORDER BY COUNT(*) DESC) AS rnk
    FROM walmart
    GROUP BY Branch, day_name
) AS ranked
WHERE rnk = 1;
```

4. Calculate the total quantity of items sold per payment method. List payment_method and total quantity.

```
SELECT payment_method, SUM(quantity) AS total_quantity_sold
FROM walmart
GROUP BY payment_method;
```

5. Determine the average. Minimum and maximum rating of category for each city.
List the city, average_rating, min_rating, and max_rating

```
SELECT
    city,
    category,
    AVG(rating) AS Average_Rating,
    MIN(rating) AS Lowest_Rating,
    MAX(rating) AS Highest_Rating
FROM walmart
GROUP BY city, category;
```

6. Determine the most common payment method for each branch. Display Branch and the preferred payment_method

```
SELECT branch, payment_method, transactions, `rank`
FROM (
    SELECT
        branch,
        payment_method,
        COUNT(*) AS transactions,
        RANK() OVER (PARTITION BY branch ORDER BY COUNT(*) DESC) AS `rank`
    FROM walmart
    GROUP BY branch, payment_method
) AS ranked
WHERE `rank` = 1
ORDER BY branch;
```

7. Identify the most active time slot for each Walmart branch based on the highest quantity of items sold.

```
SELECT branch, Time_Slot, total_quantity_sold, `rank`
FROM (
    SELECT
        branch,
        CASE
            WHEN HOUR(STR_TO_DATE(time, '%H:%i:%s')) BETWEEN 6 AND 12 THEN 'Morning'
            WHEN HOUR(STR_TO_DATE(time, '%H:%i:%s')) BETWEEN 13 AND 17 THEN 'Afternoon'
            ELSE 'Evening'
        END AS Time_Slot,
        SUM(quantity) AS total_quantity_sold,
        RANK() OVER (PARTITION BY branch ORDER BY SUM(quantity) DESC) AS `rank`
    FROM walmart
    GROUP BY branch, Time_Slot
) AS ranked
WHERE `rank` = 1;
```

8. Identify which product categories have the highest sales volume across all Walmart branches.

```
SELECT category, SUM(quantity) AS Total_Units_Sold
FROM walmart
GROUP BY category
ORDER BY Total_Units_Sold DESC;
```