Walmart DB SQL Query Report

1. Find different payment methods and show the number of transactions and the number of quantities sold for each method.

```
SELECT
    payment_method,
    COUNT(*) AS Transactions,
    SUM(quantity) AS no_of_qty
FROM walmart_db
GROUP BY payment_method;
```

2. Identify the highest-rated product category in each branch by calculating the average rating. Display the branch, category, and average rating.

```
SELECT * FROM (
    SELECT
        branch,
        category,
        AVG(rating) AS avg_rating,
        RANK() OVER(PARTITION BY branch ORDER BY AVG(rating) DESC) AS rn
    FROM walmart_db
    GROUP BY branch, category
) AS ranked_categories
WHERE rn = 1;
```

3. Determine the busiest day for each branch based on the highest number of transactions. Display the branch, day, and transaction count.

```
SELECT * FROM (
    SELECT
        branch,
        TO_CHAR(TO_DATE(date, 'DD-MM-YY'), 'Day') AS day,
        COUNT(*) AS no_of_transactions,
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rank
    FROM walmart_db
    GROUP BY branch, day
) AS ranked_days
WHERE rank = 1;
```

4. Determine the average, minimum, and maximum product ratings for each city and category. Display city, category, average rating, minimum rating, and maximum rating.

```
SELECT
    city,
    category,
    AVG(rating) AS avg_rating,
    MIN(rating) AS min_rating,
    MAX(rating) AS max_rating
FROM walmart_db
GROUP BY city, category;
```

Walmart DB SQL Query Report

5. Calculate the total profit for each product category using the formula (unit_price * quantity * profit_margin). Display category and total profit, ordered from highest to lowest.

```
SELECT
    category,
    SUM(total * profit_margin) AS total_profit
FROM walmart_db
GROUP BY category
ORDER BY total_profit DESC;
```

6. Identify the most common payment method used in each branch. Display the branch and its preferred payment method.

```
SELECT * FROM (
    SELECT
        branch,
        payment_method,
        COUNT(*) AS total_trans,
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rn
    FROM walmart_db
    GROUP BY branch, payment_method
) AS ranked_methods
WHERE rn = 1;
```

7. Categorize sales into three shifts: Morning, Afternoon, and Evening based on the time of transaction. Display branch, shift, and number of invoices.

```
SELECT
    branch,
    CASE
        WHEN EXTRACT(HOUR FROM time::time) < 12 THEN 'Morning'
        WHEN EXTRACT(HOUR FROM time::time) BETWEEN 12 AND 17 THEN 'Afternoon'
        ELSE 'Evening'
    END AS shift,
    COUNT(*) AS invoice_count
FROM walmart_db
GROUP BY branch, shift
ORDER BY branch, invoice_count DESC;
```

8. Identify the top 5 branches with the highest percentage decrease in revenue when comparing the current year (2023) to the previous year (2022).

```
WITH rev_2022 AS (
    SELECT
        branch,
        SUM(total) AS revenue
    FROM walmart_db
    WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2022
    GROUP BY branch
),
```

Walmart DB SQL Query Report

```
rev_2023 AS (
    SELECT
        branch,
        SUM(total) AS revenue
    FROM walmart_db
    WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2023
    GROUP BY branch
)
SELECT
    r22.branch,
    r22.revenue AS last_year_revenue,
    r23.revenue AS current_year_revenue,
    ROUND(((r22.revenue - r23.revenue)::NUMERIC / r22.revenue::NUMERIC) * 100, 2) AS
revenue_drop_ratio
FROM rev_2022 r22
JOIN rev_2023 r23 ON r22.branch = r23.branch
WHERE r22.revenue > r23.revenue
ORDER BY revenue_drop_ratio DESC
LIMIT 5;
```