
Walmart
Sales Analysis
END TO END PYTHON + SQL PRJOJECT

Objective

- This project is an end-to-end data analysis solution designed to extract critical business insights from Walmart sales data.
- We utilize Python for data processing and analysis, SQL for advanced querying, and structured problem-solving techniques to solve key business questions.
- The project is ideal for data analysts looking to develop skills in data manipulation, SQL querying, and data pipeline creation

Q1: Find different payment methods, number of transactions, and quantity sold by payment method

```sql
select payment_method ,
count(payment_method) as 'no_of_transactions',
sum(quantity) as 'total_quantity'
from walmart
group by payment_method;
```

Q2: Calculate the total quantity of items sold per payment method

```sql
select payment_method ,
sum(quantity) as 'total_quantity'
from walmart
group by payment_method;
```

Q3:Identify the highest-rated category in each branch
Display the branch, category, and avg rating

```sql
SELECT branch, category, avg_rating
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;
```

Q4: Identify the busiest day for each branch based on the number of transactions

```sql
SELECT branch, day_name, no_transactions
FROM (
    SELECT
        branch,
        DAYNAME(STR_TO_DATE(date, '%d/%m/%Y')) AS day_name,
        COUNT(*) AS no_transactions,
        DENSE_RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS `rank`
    FROM walmart
    GROUP BY branch, DAYNAME(STR_TO_DATE(date, '%d/%m/%Y'))
) AS ranked
WHERE `rank` = 1;
```

Q5: Determine the average, minimum, and maximum rating of categories for each city

```sql
select City , category,
avg(rating) as 'avg_rating',
min(rating) as 'min_rating',
max(rating) as 'max_rating'
from walmart
group by  City ,category;
```

Q6: Calculate the total profit for each category

```sql
select category,
sum(unit_price * quantity * profit_margin) as 'total_profit'
from walmart
group by category;
```

Q7: Determine the most common payment method for each branch

```sql
WITH cte AS (
    SELECT
        branch,
        payment_method,
        COUNT(*) AS total_trans,
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS `rank`
    FROM walmart
    GROUP BY branch, payment_method
)
SELECT branch, payment_method AS preferred_payment_method
FROM cte
WHERE `rank` = 1;
```

Q8: Categorize sales into Morning, Afternoon, and Evening shifts

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