**COFFEE SHOP SALES DASHBOARD QUERIES**

1. Total Sales for a specific month

SELECT ROUND(SUM(unit\_price \* transaction\_qty)) as Total\_Sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5; -- for month of May



1. Total Sales Difference and MOM growth

WITH monthly\_sales AS (

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

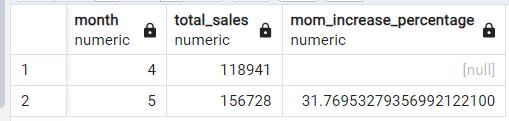
total\_sales,

(total\_sales - LAG(total\_sales) OVER (ORDER BY month)) /

LAG(total\_sales) OVER (ORDER BY month) \* 100 AS mom\_increase\_percentage

FROM

monthly\_sales;

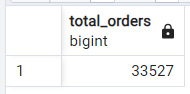


1. Total orders for a specific month

SELECT COUNT(transaction\_id) as Total\_Orders

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5; -- for month of May



1. Total Orders Difference and MOM growth

WITH monthly\_orders AS (

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

COUNT(transaction\_id) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5) -- for April and May

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

total\_orders,

CASE

WHEN LAG(total\_orders) OVER (ORDER BY month) <> 0 THEN

ROUND((total\_orders - LAG(total\_orders) OVER (ORDER BY month))::NUMERIC /

LAG(total\_orders) OVER (ORDER BY month) \* 100, 2)

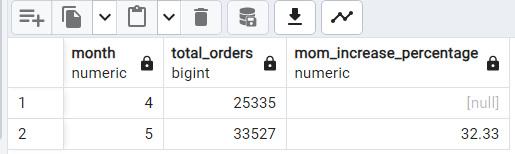
ELSE

NULL

END AS mom\_increase\_percentage

FROM

monthly\_orders;

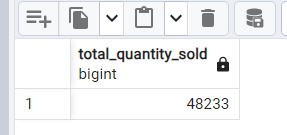


1. Total Quantity Sold for a month

SELECT SUM(transaction\_qty) as Total\_Quantity\_Sold

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5; -- for month of May



1. Total Quantity sold Difference and MOM growth

WITH monthly\_orders AS (

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

SUM(transaction\_qty) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5) -- for April and May

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

total\_orders,

CASE

WHEN LAG(total\_orders) OVER (ORDER BY month) <> 0 THEN

ROUND((total\_orders - LAG(total\_orders) OVER (ORDER BY month))::NUMERIC /

LAG(total\_orders) OVER (ORDER BY month) \* 100, 2)

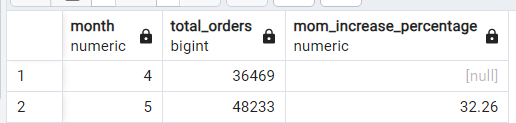
ELSE

NULL

END AS mom\_increase\_percentage

FROM

monthly\_orders;



1. Day wise total sales, total quantity sold, total orders

SELECT

SUM(unit\_price \* transaction\_qty) AS total\_sales,

SUM(transaction\_qty) AS total\_quantity\_sold,

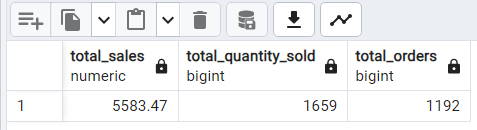
COUNT(transaction\_id) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

transaction\_date = '2023-05-18'; -- For 18 May 2023



1. Dailly sales month wise

SELECT

EXTRACT(DAY FROM transaction\_date) AS day\_of\_month,

ROUND(SUM(unit\_price \* transaction\_qty), 1) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

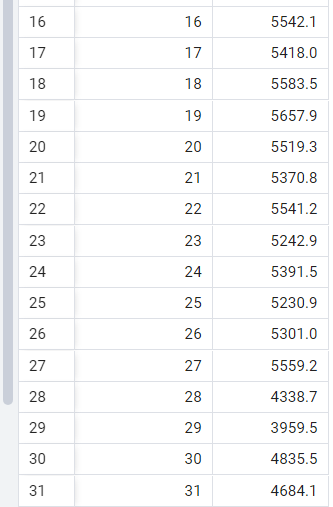
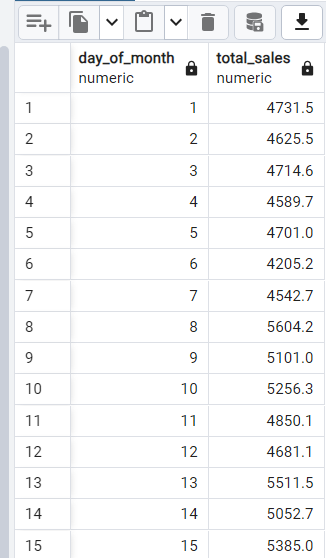
EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

EXTRACT(DAY FROM transaction\_date)

ORDER BY

EXTRACT(DAY FROM transaction\_date);



1. Daily avg Sales is below / above average

SELECT

day\_of\_month,

CASE

WHEN total\_sales > avg\_sales THEN 'Above Average'

WHEN total\_sales < avg\_sales THEN 'Below Average'

ELSE 'Average'

END AS sales\_status,

total\_sales

FROM (

SELECT

EXTRACT(DAY FROM transaction\_date) AS day\_of\_month,

SUM(unit\_price \* transaction\_qty) AS total\_sales,

AVG(SUM(unit\_price \* transaction\_qty)) OVER () AS avg\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

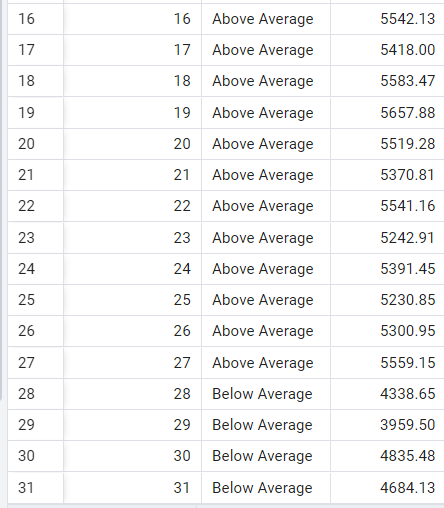
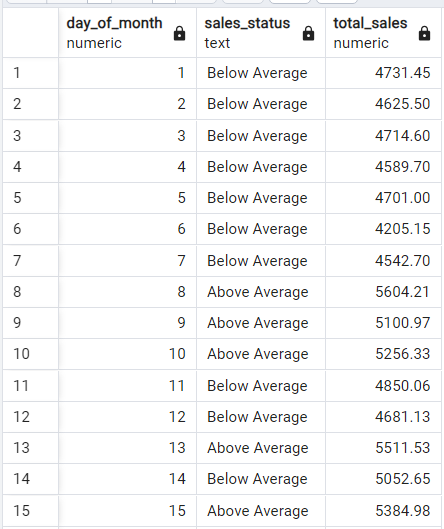
GROUP BY

EXTRACT(DAY FROM transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;



1. Sales by weekdays / weekends

SELECT

CASE

WHEN EXTRACT(DOW FROM transaction\_date) IN (0, 6) THEN 'Weekends'

ELSE 'Weekdays'

END AS day\_type,

ROUND(SUM(unit\_price \* transaction\_qty), 2) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

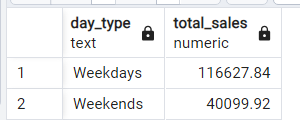
GROUP BY

CASE

WHEN EXTRACT(DOW FROM transaction\_date) IN (0, 6) THEN 'Weekends'

ELSE 'Weekdays'

END;



1. Sales by store location

SELECT

store\_location,

SUM(unit\_price \* transaction\_qty) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

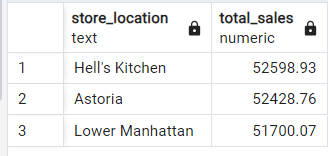
EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

store\_location

ORDER BY

SUM(unit\_price \* transaction\_qty) DESC;



1. Sales by Product Category

SELECT

product\_category,

ROUND(SUM(unit\_price \* transaction\_qty), 1) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

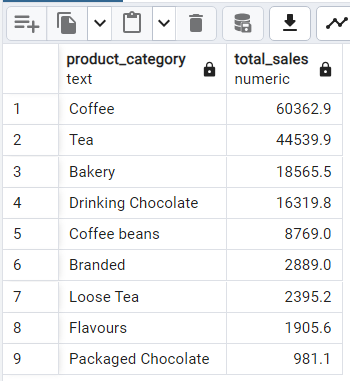
EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

product\_category

ORDER BY

SUM(unit\_price \* transaction\_qty) DESC;



1. Sales by top products (top 10)

SELECT

product\_type,

ROUND(SUM(unit\_price \* transaction\_qty), 1) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

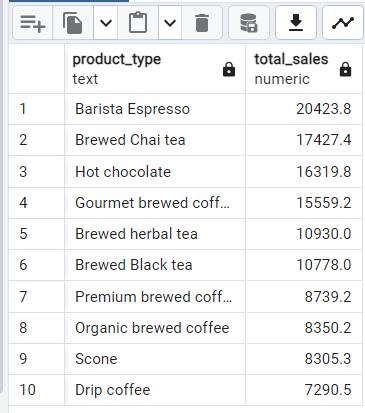
GROUP BY

product\_type

ORDER BY

SUM(unit\_price \* transaction\_qty) DESC

LIMIT 10;



1. Sales by Day / hour

SELECT

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales,

SUM(transaction\_qty) AS total\_quantity,

COUNT(\*) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

EXTRACT(DOW FROM transaction\_date) = 2 -- Filter for Tuesday (0 is Sunday, 1 is Monday, ..., 6 is Saturday)

AND EXTRACT(HOUR FROM transaction\_time) = 8 -- Filter for hour number 8

AND EXTRACT(MONTH FROM transaction\_date) = 5; -- Filter for May (month number 5)



1. Day wise sales

SELECT

CASE

WHEN EXTRACT(DOW FROM transaction\_date) = 1 THEN 'Monday'

WHEN EXTRACT(DOW FROM transaction\_date) = 2 THEN 'Tuesday'

WHEN EXTRACT(DOW FROM transaction\_date) = 3 THEN 'Wednesday'

WHEN EXTRACT(DOW FROM transaction\_date) = 4 THEN 'Thursday'

WHEN EXTRACT(DOW FROM transaction\_date) = 5 THEN 'Friday'

WHEN EXTRACT(DOW FROM transaction\_date) = 6 THEN 'Saturday'

ELSE 'Sunday'

END AS day\_of\_week,

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

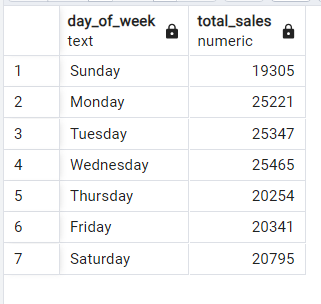
EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

EXTRACT(DOW FROM transaction\_date) -- Group by the day of the week

ORDER BY

EXTRACT(DOW FROM transaction\_date);



1. Hour wise sales

SELECT

EXTRACT(HOUR FROM transaction\_time) AS hour\_of\_day,

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

EXTRACT(HOUR FROM transaction\_time)

ORDER BY

EXTRACT(HOUR FROM transaction\_time);

