## Creating table for the data

create table coffee\_shop\_sales

(

transaction\_id int,

transaction\_date date,

transaction\_time time,

transaction\_qty int,

store\_id int,

store\_location varchar(30),

product\_id int,

unit\_price decimal,

product\_category varchar(25),

product\_type varchar(50),

product\_detail varchar(50)

)

### Importing the csv file

copy coffee\_shop\_sales

from 'C:\Program Files\PostgreSQL\16\data\data\_copy\coffee\_shop\_sales.csv'

delimiter ','

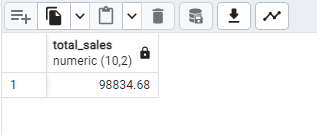
csv header;

## Finding total sales of the month- here 3 = March

SELECT CAST(ROUND(SUM(unit\_price \* transaction\_qty), 2) AS NUMERIC(10, 2)) AS Total\_Sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 3;



## Finding month-on-month increase or decrease in sales

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month\_sale,

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

ROUND(

(

(

SUM(unit\_price \* transaction\_qty) - LAG(SUM(unit\_price \* transaction\_qty), 1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date))

) / LAG(SUM(unit\_price \* transaction\_qty), 1) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date))

) \* 100,

2

) AS mon\_increase\_percentage

FROM

coffee\_shop\_sales

WHERE

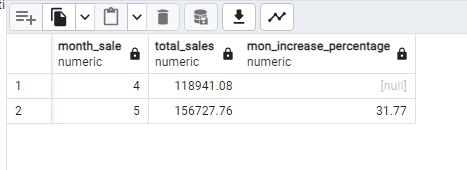
EXTRACT(MONTH FROM transaction\_date) IN (4, 5)

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



## Finding all months sales, January to June

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month\_sale,

EXTRACT(YEAR FROM transaction\_date) AS year\_sale,

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

FROM

coffee\_shop\_sales

GROUP BY

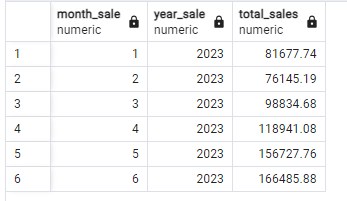
EXTRACT(YEAR FROM transaction\_date),

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(YEAR FROM transaction\_date),

EXTRACT(MONTH FROM transaction\_date);



### another code with month name

SELECT

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

ELSE 'Unknown' -- Handle unexpected cases

END AS month\_name,

EXTRACT(YEAR FROM transaction\_date) AS year\_sale,

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

FROM

coffee\_shop\_sales

GROUP BY

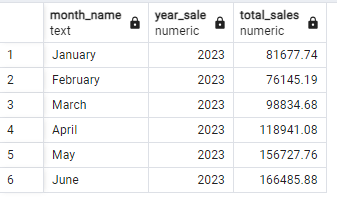
EXTRACT(YEAR FROM transaction\_date), -- Group by year

EXTRACT(MONTH FROM transaction\_date) -- Group by month

ORDER BY

EXTRACT(YEAR FROM transaction\_date),

EXTRACT(MONTH FROM transaction\_date);



## Finding the difference in sales between the selected month and the previous month.

WITH MonthlySales AS (

SELECT

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

ELSE 'Unknown'

END AS month\_sale,

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

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

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)) AS prev\_month\_sales

FROM

coffee\_shop\_sales

WHERE

EXTRACT(YEAR FROM transaction\_date) = 2023

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month\_sale,

total\_sales,

COALESCE(total\_sales - prev\_month\_sales, 0) AS sales\_difference

FROM

MonthlySales

ORDER BY

CASE month\_sale

WHEN 'January' THEN 1

WHEN 'February' THEN 2

WHEN 'March' THEN 3

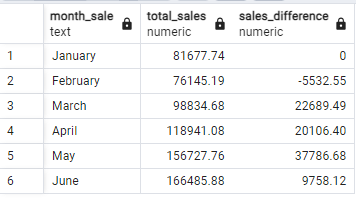
WHEN 'April' THEN 4

WHEN 'May' THEN 5

WHEN 'June' THEN 6

ELSE 7 -- 'Unknown' will appear at the end

END;



## calculate the total number of orders for each month

select \* from coffee\_shop\_sales

SELECT

count(transaction\_id) AS total\_qty,

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

ELSE 'Unknown' -- Handle unexpected cases

END AS month\_name

FROM

coffee\_shop\_sales

GROUP BY

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

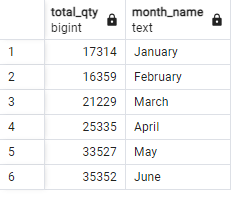
WHEN 12 THEN 'December'

ELSE 'Unknown'

END

ORDER BY

MIN(EXTRACT(MONTH FROM transaction\_date)); -- Optional: Order by the numerical month order



## Determine the month-on-month increase or decrease in the number of orders

SELECT

EXTRACT(MONTH FROM transaction\_date) AS MONTH,

CAST(ROUND(COUNT(transaction\_id)) AS INTEGER) AS total\_orders,

ROUND(

(

(COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)))::NUMERIC /

LAG(COUNT(transaction\_id), 1) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date))

) \* 100,

2

) AS month\_increase\_percentage

FROM

coffee\_shop\_sales

WHERE

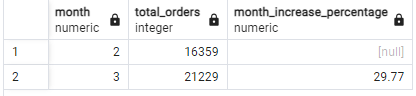
EXTRACT(MONTH FROM transaction\_date) IN (2, 3)

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



## Calculate the total quantity sold for every month

SELECT

SUM(transaction\_qty) AS total\_quantity,

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

ELSE 'Unknown'

END AS month\_name

FROM

coffee\_shop\_sales

WHERE

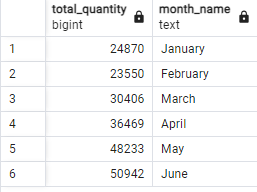
EXTRACT(YEAR FROM transaction\_date) = 2023

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



## Total quantity sold on month difference and month growth

SELECT

month\_name,

total\_quantity,

ROUND(

(total\_quantity - LAG(total\_quantity, 1) OVER (ORDER BY month\_num))::numeric

/ NULLIF(LAG(total\_quantity, 1) OVER (ORDER BY month\_num), 0) \* 100

, 2) AS sales\_diff

FROM (

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month\_num,

CASE EXTRACT(MONTH FROM transaction\_date)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

ELSE 'Unknown'

END AS month\_name,

SUM(transaction\_qty) AS total\_quantity

FROM

coffee\_shop\_sales

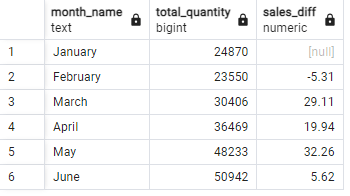
GROUP BY

EXTRACT(MONTH FROM transaction\_date)

) AS monthly\_totals

ORDER BY

month\_num;



## Calculate total sum, total quantity sold and total orders for particular date

select

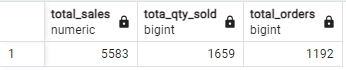
round(sum(unit\_price \* transaction\_qty)) as total\_sales,

sum (transaction\_qty) as tota\_qty\_sold,

count ( transaction\_id) as total\_orders

from coffee\_shop\_sales

where transaction\_date = '2023-05-18'



SELECT

CONCAT(

ROUND(SUM(unit\_price \* transaction\_qty) / 1000.0, 1),

'k'

) AS total\_sales\_display,

CONCAT(

ROUND(SUM(transaction\_qty) / 1000.0, 1),

'k'

) AS total\_qty\_sold,

CONCAT(

ROUND(COUNT(transaction\_id) / 1000.0, 1),

'k'

) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

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



## find sales on weekend

SELECT

CASE

WHEN EXTRACT(DOW FROM transaction\_Date) IN (0, 6) THEN 'WEEKENDS'

ELSE 'Weekdays'

END AS day\_sales,

concat(round(SUM(transaction\_qty \* unit\_price)/1000,1), 'k') AS total\_sales

FROM

coffee\_shop\_sales

where extract(month from transaction\_date) = 2

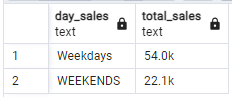
GROUP BY

CASE

WHEN EXTRACT(DOW FROM transaction\_Date) IN (0, 6) THEN 'WEEKENDS'

ELSE 'Weekdays'

END;



## SALES VALUE BY STORE LOCATION

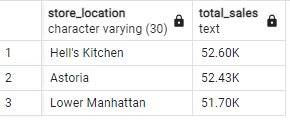
SELECT store\_location, concat(round(sum(unit\_price\*transaction\_qty)/1000,2), 'K') as total\_sales

from coffee\_shop\_sales

where extract(month from transaction\_date)=5

group by store\_location

order by sum(unit\_price\*transaction\_qty) desc;



## Daily sale for particular month

SELECT

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

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

FROM

coffee\_shop\_sales

WHERE

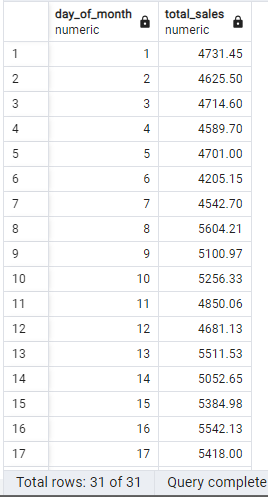
EXTRACT(MONTH FROM transaction\_date) = 5 -- Filter for May

GROUP BY

EXTRACT(DAY FROM transaction\_Date) -- Group by day of the month

ORDER BY

EXTRACT(DAY FROM transaction\_Date); -- Optional: Order by day of the month



## Calculating total\_sales and avg sales for everyday and finding avg line

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

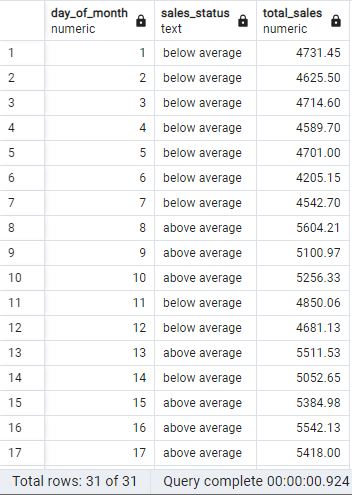
GROUP BY

extract(day from transaction\_date)

) AS sales\_summary

ORDER BY

day\_of\_month;



## Calculate sales by product category

select product\_category,

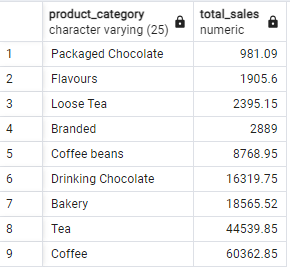
sum(transaction\_qty\* unit\_price) as total\_Sales

from coffee\_shop\_sales

where extract(month from transaction\_date)=5

group by product\_Category

order by 2;



## Finding top 10 product by sales

select product\_type,

sum(transaction\_qty\* unit\_price) as total\_Sales

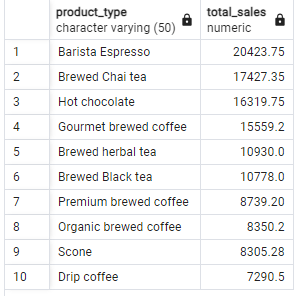
from coffee\_shop\_sales

where extract(month from transaction\_date)=5

group by product\_type

order by 2 desc

limit 10;



## Finding total sales, total quantity sold in particular time of the day in particular month

SELECT

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

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

COUNT(\*)

FROM

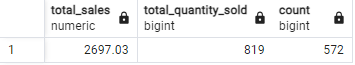
coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5

AND EXTRACT(DOW FROM transaction\_date) = 1

AND EXTRACT(HOUR FROM transaction\_time) = 8



## Find total sales in the hour of the day in month

select extract(hour from transaction\_time),

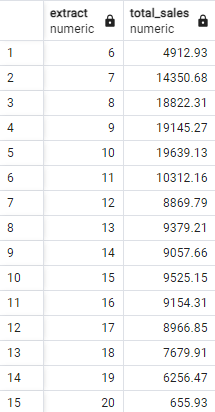
sum(unit\_price\*transaction\_qty) as total\_sales

from coffee\_shop\_sales

where extract(month from transaction\_date)=5

group by extract(hour from transaction\_time)

order by extract(hour from transaction\_time)



## Calculate total sale for per day of the month

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 days\_of\_the\_week,

round(sum(unit\_price\*transaction\_qty)) as total\_sales

from

coffee\_shop\_sales

where extract(month from transaction\_date)=5

group by

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;

