**COFFEE SHOP – DATA ANALYSIS PROJECT**

**POSTGRESQL – QUERIES**

**IMPORT FILE**

1. Download the Dataset as extension. xlsv
2. Convert it into the format of .csv
3. Create the Table with name ‘coffee\_shop\_sales’.
4. Import the file(Dataset) by right clicking the table and select the option of “Import/Export” from local device.
5. Check the dataset after imported it.

**KPI REQUIREMENTS**

**TOTAL SALES ANALYSIS**

**Total Sales**

SELECT ROUND(SUM(unit\_price \* transaction\_qty / 1000.0)) || ' K' AS total\_sales

FROM coffee\_shop\_sales

**Total Sales by Specific Month**

SELECT ROUND(SUM(unit\_price \* transaction\_qty / 1000.0)) || ' K' AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5; --May Month

**Difference the total sales in between selected month and previous month**

WITH monthly\_sales AS -- CTE for ease calculations

(

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

SUM(unit\_price \* transaction\_qty) / 1000.0 AS total\_sales\_k

FROM

coffee\_shop\_sales

WHERE

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

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

ROUND(total\_sales\_k) || ' K' AS total\_sales,

ROUND(

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

/ NULLIF(LAG(total\_sales\_k) OVER (ORDER BY month), 0) \* 100, 2 ) AS mom\_increase\_percentage

FROM monthly\_sales

ORDER BY month;

**TOTAL ORDERS ANALYSIS**

**Total Orders**

SELECT COUNT(\*) FROM coffee\_shop\_sales;

**Total Orders by Specific Month**

SELECT COUNT(\*) AS total\_orders

FROM coffee\_shop\_sales

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

**Difference the total orders in between selected month and previous month**

WITH monthly\_orders AS

(

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

COUNT(\*) AS total\_orders\_m

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (3, 4)

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

total\_orders\_m AS total\_orders,

ROUND(

(total\_orders\_m - LAG(total\_orders\_m) OVER (ORDER BY month))

/ NULLIF(LAG(1.0 \* total\_orders\_m) OVER (ORDER BY month), 0) \* 100, 2)

AS mom\_increase\_percentage

FROM monthly\_orders

ORDER BY month;

**TOTAL QUANTITY ANALYSIS**

**Total Quantity**

SELECT SUM(transaction\_qty) FROM coffee\_shop\_sales;

**Total Quantity for specific month**

SELECT SUM(transaction\_qty)

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 1; -- Jan Quantity

**Difference the total quantity in between selected month and previous month**

WITH month\_qty AS

(

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

SUM(transaction\_qty) AS total\_qty\_m

FROM coffee\_shop\_sales

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

GROUP BY EXTRACT(MONTH FROM transaction\_date)

)

SELECT

month,

total\_qty\_m AS total\_qty,

ROUND(

(total\_qty\_m – LAG(total\_qty\_m) OVER (ORDER BY month))

/ NULLIF(1.0 \* LAG(total\_qty\_m) OVER (ORDER BY month), 0) \* 100,2) AS mom\_increase\_percentage

FROM month\_qty

ORDER BY month;

**CHART REQUIREMENTS**

**Total Sales, Orders and Quantity for selected date**

SELECT

ROUND(1.0\*COUNT(transaction\_id)/1000,1) || ' K' AS total\_orders,

ROUND(1.0\*SUM(transaction\_qty)/1000,1) || ' K' AS total\_qty\_aold,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales

FROM

coffee\_shop\_sales

WHERE DATE(transaction\_date) = '2023-03-27';

**Weekday and Weekend Sales**

SELECT

CASE

WHEN EXTRACT(DOW FROM transaction\_date) IN (0, 6) THEN 'Weekend' -- 0 = Sunday, 6 = Saturday

ELSE 'Weekday'

END AS day\_type,

ROUND(SUM(unit\_price \* transaction\_qty) / 1000, 1) || ' K' AS total\_sales\_k

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 3

GROUP BY

day\_type;

**Sales by store location for specified month**

SELECT

store\_location,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ‘ K’ AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 6

GROUP BY store\_location

ORDER BY total\_sales DESC;

**Sales by store location for specified month – WITH Clause**

WITH daily\_sales

AS

(

SELECT

transaction\_date,

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

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5

GROUP BY transaction\_date

)

SELECT

ROUND(AVG(total\_sales), 2) AS avg\_sales\_per\_day

FROM

daily\_sales;

**Sales by store location for specified month – INNER QUERY Method**

SELECT

ROUND(AVG(total\_sales)/1000,1) || ‘ K’ AS avg\_sales

FROM

(

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

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 2

GROUP BY transaction\_date

) AS inner\_query;

**Daywise Sales for specified month**

SELECT

EXTRACT(DAY FROM transaction\_date) AS date,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales\_per\_day

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 2

GROUP BY date

ORDER BY date ASC;

**Daywise Sales for specified month and compare with average sales**

WITH daily\_sales AS

(

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

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 2

GROUP BY EXTRACT(DAY FROM transaction\_date)

),

average\_sales AS

(

SELECT AVG(total\_sales) AS avg\_sales

FROM daily\_sales

)

SELECT

ds.day\_of\_month,

CASE

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

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

ELSE 'Equal to Average'

END AS sales\_status,

ds.total\_sales || 'K' AS total\_sales

FROM

daily\_sales AS ds,

average\_sales AS avg

ORDER BY

ds.day\_of\_month;

**Total Sales by product category**

SELECT

product\_category,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 3

GROUP BY product\_category

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

**Top 10 sales by product type**

SELECT

product\_type,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 3

GROUP BY product\_type

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

LIMIT 10;

**Top 10 sales by product type with specific product category**

SELECT

product\_type,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 3 AND product\_category = 'Tea'

GROUP BY product\_type

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

LIMIT 10;

**Total Sales, Quantity and Orders due to the month, day of week and transaction time**

SELECT

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales,

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

COUNT(\*) AS total\_orders

FROM coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) = 5

AND EXTRACT(DOW FROM transaction\_date) = 0

AND EXTRACT(HOUR FROM transaction\_time)= 14;

**Peak hour of sales on specific month**

SELECT

EXTRACT(HOUR FROM transaction\_time) AS hour,

ROUND(SUM(unit\_price \* transaction\_qty)/1000,1) || ' K' AS total\_sales

FROM cofFee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5

GROUP BY EXTRACT(HOUR FROM transaction\_time)

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