**Coffee Shop Sales Dashboard SQL Queries**

1. Display the entire table with ascending order of transaction\_id

SELECT \*

FROM [Coffee Shop Sales]

ORDER BY transaction\_id

1. Total Sales Analysis:
2. Calculate Total Sales for each respective month(with 1 decimal place) –

SELECT ROUND(SUM(transaction\_qty \* unit\_price), 1) AS Total\_Sales

FROM [Coffee Shop Sales]

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

1. Calculate Mom increase/decrease and calculate the difference in Sales between selected month and previous month -

SELECT

MONTH(transaction\_date) AS Month, --Month Number

ROUND(SUM(transaction\_qty \* unit\_price)) AS Total\_Sales, --Total Sales Column

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

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

OVER(ORDER BY MONTH(transaction\_date))\*100 AS MOM\_Increase\_Percentage

FROM [Coffee Shop Sales]

WHERE MONTH(transaction\_date) IN(4,5) – For months of April and May

GROUP BY MONTH(transaction\_date)

ORDER BY MONTH(transaction\_date)

1. Total Order Analysis:
2. Calculate Total Orders placed for each respective month –

SELECT  COUNT(transaction\_id) AS Total\_Orders

FROM [Coffee Shop Sales]

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

1. Calculate Mom increase/decrease and calculate the difference in orders placed between selected month and previous month -

SELECT

    MONTH(transaction\_date) AS month,

    ROUND(COUNT(transaction\_id), 1) AS total\_orders,

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

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

    OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS MOM\_increase\_percentage

    FROM [Coffee Shop Sales]

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

    GROUP BY MONTH(transaction\_date)

    ORDER BY MONTH(transaction\_date);

1. Total Quantity Analysis:
2. Calculate Total quantity sold for each respective month –

SELECT

    SUM(transaction\_qty) AS Total\_Quantity\_Sold

    FROM [Coffee Shop Sales]

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

1. Calculate Mom increase/decrease and calculate the difference in orders placed between selected month and previous month -

SELECT

MONTH(transaction\_date) AS Month,

ROUND(SUM(transaction\_qty), 1) AS Total\_Quantity\_Sold,

SUM(transaction\_qty) – LAG(SUM(transaction\_qty), 1)

OVER (ORDER BY MONTH(transaction\_date))) /LAG(SUM(transaction\_qty), 1)

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS MOM\_Increase\_Percentage

FROM Coffee Shop Sales

WHERE MONTH(transaction\_date) IN (4,5) – for April and May

GROUP BY MONTH(transaction\_date)

ORDER BY MONTH(transaction\_date);

1. Displaying the metrics for the daily Sales, Quantity Sold and Orders Placed, we will use the following query:
2. 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. For rounded off values, use the below query:

SELECT

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

  CONCAT(ROUND(SUM(transaction\_qty)/1000, 1), 'K') AS total\_quantity\_sold,

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

  FROM [Coffee Shop Sales]

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

1. The Sales Analysis for the store wise location in descending order of Total Sales –

SELECT store\_location,

SUM(unit\_price \* transaction\_qty) AS Total\_Sales

FROM [Coffee Shop Sales]

WHERE MONTH(transaction\_date) = 5

GROUP BY store\_location

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

1. Now, for the daily Sales Analysis with Average Line, we’ll start with the following query –

SELECT AVG(Total\_Sales) AS Average\_Sales

FROM

(

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

FROM [Coffee Shop Sales]

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

GROUP BY transaction\_date – for Individual dates

) AS Internal\_Query

1. Next step is calculating the daily sales for a selected month using the below query –

SELECT

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

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

FROM

[Coffee Shop Sales]

WHERE

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

GROUP BY

DAY(transaction\_date)

ORDER BY

DAY(transaction\_date);

1. Sales Trend for the daily sales with Sales status for each day(whether it's Below Average and 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

        DAY(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

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

        GROUP BY

        DAY(transaction\_date)

      )AS sales\_data

      ORDER BY

      day\_of\_month;

1. For analysing the Sales w.r.t. Product Category, we will use the below query -

      SELECT

      product\_category,

      ROUND(SUM(unit\_price \* transaction\_qty),1) as Total\_Sales

      FROM [Coffee Shop Sales]

      WHERE

      MONTH(transaction\_date) = 5

      GROUP BY product\_category

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

1. The query for examining the Top 10 Product Type is fired as follows -

      SELECT

      TOP 10 product\_type,

      ROUND(SUM(unit\_price \* transaction\_qty),1) as Total\_Sales

      FROM [Coffee Shop Sales]

      WHERE

      MONTH(transaction\_date) = 5

      GROUP BY product\_type

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

1. In the Sales analysis by days and hours for the Coffee Shop -

      SELECT

      CASE

        WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

        WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

        WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

        WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

        WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

        WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

        ELSE 'Sunday'

        END AS Day\_of\_Week,

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

        FROM [Coffee Shop Sales]

        WHERE

        MONTH(transaction\_date) = 5 -- Filter for May (month number 5)

        GROUP BY

        CASE

        WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

        WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

        WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

        WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

        WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

        WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

        ELSE 'Sunday'S

        END;

1. TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

    SELECT

    HOUR(transaction\_time) AS Hour\_of\_Day,

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

    FROM

    [Coffee Shop Sales]

    WHERE

    MONTH(transaction\_date) = 5 -- Filter for May (month number 5)

    GROUP BY

    HOUR(transaction\_time)

    ORDER BY

    HOUR(transaction\_time);