--1. Total Sales Analysis:

------ Total\_Sales For 1 Month

SELECT

MONTHNAME(transaction\_date) as Month\_Name,

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

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

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

GROUP BY

MONTHNAME(transaction\_date)

------ Total\_sales For all 6 Months with Month name

SELECT

MONTH(transaction\_date) as month\_number,

MONTHNAME(transaction\_date) as month\_name,

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

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

GROUP BY

MONTH(transaction\_date), MONTHNAME(transaction\_date)

ORDER BY

month\_number;

-------Determine the month-on-month increase or decrease in the number of Sales

SELECT

MONTH(transaction\_date) AS month\_number,

MONTHNAME(transaction\_date) AS month\_name,

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

CASE

WHEN LAG(SUM(unit\_price \* transaction\_qty), 1) OVER (ORDER BY MONTH(transaction\_date)) = 0 THEN NULL

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

LAG(SUM(unit\_price \* transaction\_qty), 1) OVER (ORDER BY MONTH(transaction\_date))) \* 100

END AS mom\_increase\_percentage

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) IN (2, 1) -- for months of April and May

GROUP BY

MONTH(transaction\_date), MONTHNAME(transaction\_date)

ORDER BY

MONTH(transaction\_date);

--2. Total Orders Analysis:

----- Total\_Orders For all Months in table

SELECT

MONTH(transaction\_date) as month\_number,

MONTHNAME(transaction\_date) as month\_name,

COUNT(transaction\_id) as Total\_Orders

FROM COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

GROUP BY

MONTH(transaction\_date), MONTHNAME(transaction\_date)

ORDER BY

month\_number;

-- Total\_Orders For perticular month or months with month name ;

SELECT

MONTH(transaction\_date) as month\_number,

MONTHNAME(transaction\_date) as month\_name,

COUNT(transaction\_id) as Total\_Orders

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) IN (1) -- for months of jan to june

GROUP BY

month\_name, month\_number

ORDER BY

month\_number;

----TOTAL ORDERS For 1 Month

SELECT COUNT(transaction\_id) as Total\_Orders

FROM COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

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

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

SELECT

MONTH(transaction\_date) AS month,

MONTHNAME(transaction\_date) AS month\_name,

ROUND(COUNT(transaction\_id)) 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.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) IN (4,5) -- for Jan to June

GROUP BY

MONTH(transaction\_date), MONTHNAME(transaction\_date)

ORDER BY

MONTH(transaction\_date);

--3. Total Quantity Sold Analysis:

--DAILY SALES, QUANTITY and TOTAL ORDERS For spefic day

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.PUBLIC.COFFEE\_SALES

WHERE

transaction\_date = '2023-01-24';

--DAILY SALES, QUANTITY and TOTAL ORDERS For every day

SELECT

transaction\_date,

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.PUBLIC.COFFEE\_SALES

GROUP BY

transaction\_date

ORDER BY

transaction\_date;

--If you want to get exact Rounded off values then use below query to get the result:

SELECT

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

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

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

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

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

--SALES TREND OVER PERIOD (AVERAGE)

SELECT

MONTH(transaction\_date) as month\_number,

MONTHNAME(transaction\_date) as month\_name,

AVG(total\_sales) AS average\_sales

FROM (

SELECT

transaction\_date,

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

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

GROUP BY

transaction\_date

) AS internal\_query

GROUP BY

MONTH(transaction\_date), MONTHNAME(transaction\_date)

ORDER BY

MONTH(transaction\_date);

--COMPARING DAILY SALES WITH AVERAGE SALES – IF GREATER THAN “ABOVE AVERAGE” and LESSER THAN “BELOW 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.PUBLIC.COFFEE\_SALES

WHERE

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

GROUP BY

DAY(transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;

--SALES BY WEEKDAY / WEEKEND:

WITH DayTypeCTE AS (

SELECT

CASE

WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

ELSE 'Weekdays'

END AS day\_type,

unit\_price \* transaction\_qty AS sales,

transaction\_qty

FROM

COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

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

)

SELECT

day\_type,

ROUND(SUM(sales), 2) AS total\_sales,

SUM(transaction\_qty) AS Total\_Qty

FROM

DayTypeCTE

GROUP BY

day\_type;

--SALES BY STORE LOCATION

SELECT

store\_location,

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

SUM(transaction\_qty) as Total\_Qty

FROM COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) = 5

GROUP BY store\_location

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

--SALES BY PRODUCT CATEGORY per month

SELECT

product\_category,

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

FROM COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) In (2)

GROUP BY product\_category

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

--SALES BY PRODUCTS (TOP 10) Per Month

SELECT

product\_type,

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

FROM COFFEE\_SHOP\_SALES.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) In (1)

GROUP BY product\_type

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

LIMIT 10

--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.PUBLIC.COFFEE\_SALES

WHERE

DAYOFWEEK(transaction\_date) = 1 -- Filter for Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)

AND HOUR(transaction\_time) = 2 -- Filter for hour number 8

AND MONTH(transaction\_date) = 1; -- Filter for May (month number 5)

--TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY

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.PUBLIC.COFFEE\_SALES

WHERE

MONTH(transaction\_date) = 1 -- 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'

END;

--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.PUBLIC.COFFEE\_SALES

WHERE

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

GROUP BY

HOUR(transaction\_time)

ORDER BY

HOUR(transaction\_time);