USE coffee\_shop\_sales\_db; -- Switch to the coffee\_shop\_sales\_db database

GO

-- Retrieve all records from the coffee\_shop\_sales table

SELECT \* FROM coffee\_shop\_sales;

-- Total Sales for May

SELECT

ROUND(SUM(unit\_price \* transaction\_qty), 2) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5;

-- Monthly Sales Comparison for April and May

SELECT

MONTH(transaction\_date) AS month,

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

(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 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);

-- Total Orders for May

SELECT

COUNT(transaction\_id) as Total\_Orders

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5;

-- Monthly Orders Comparison for April and May

SELECT

MONTH(transaction\_date) AS month,

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

WHERE

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

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

-- Total Quantity Sold for May

SELECT

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

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5;

-- Monthly Quantity Comparison for April and May

SELECT

MONTH(transaction\_date) AS month,

ROUND(SUM(transaction\_qty), 2) 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;

-- Sales Data for 18th May 2023

SELECT

ROUND(SUM(unit\_price \* transaction\_qty) / 1000, 1) 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';

-- Sales Data for Weekends and Weekdays on 18th May 2023

SELECT

CASE

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

ELSE 'Weekdays'

END AS day\_type,

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

GROUP BY

CASE

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

ELSE 'Weekdays'

END;

-- Sales by Store Location for May

SELECT

store\_location,

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

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

store\_location

ORDER BY

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

-- Average Sales for May

SELECT

AVG(unit\_price \* transaction\_qty) AS Avg\_sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5;

-- Average Daily Sales for May

SELECT

ROUND(AVG(total\_sales), 1) AS Avg\_Sales

FROM

(

SELECT

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

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

transaction\_date

) AS Internal\_query;

-- Daily Sales for May

SELECT

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

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

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

DAY(transaction\_date)

ORDER BY

DAY(transaction\_date);

-- Sales Status Compared to Average for May

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

GROUP BY

DAY(transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;

-- Sales by Product Category for May

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;

-- Top 10 Products by Sales for May

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;

-- Sales Analysis for Tuesday at 8 AM in May

SELECT

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

SUM(transaction\_qty) AS Total\_Quantity,

COUNT(\*) AS Total\_orders

FROM

coffee\_shop\_sales

WHERE

DATEPART(WEEKDAY, transaction\_date) = 3 -- Filter for Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)

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

AND MONTH(transaction\_date) = 5;

-- Sales by Hour for May

SELECT

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

ROUND(SUM(unit\_price \* transaction\_qty), 0) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

DATEPART(HOUR, transaction\_time)

ORDER BY

Hour\_of\_Day;

-- Sales by Day of the Week for May

WITH SalesByDay AS (

SELECT

CASE

WHEN DATEPART(WEEKDAY, transaction\_date) = 2 THEN 'Monday'

WHEN DATEPART(WEEKDAY, transaction\_date) = 3 THEN 'Tuesday'

WHEN DATEPART(WEEKDAY, transaction\_date) = 4 THEN 'Wednesday'

WHEN DATEPART(WEEKDAY, transaction\_date) = 5 THEN 'Thursday'

WHEN DATEPART(WEEKDAY, transaction\_date) = 6 THEN 'Friday'

WHEN DATEPART(WEEKDAY, transaction\_date) = 7 THEN 'Saturday'

ELSE 'Sunday'

END AS Day\_of\_Week,

unit\_price \* transaction\_qty AS Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

)

SELECT

Day\_of\_Week,

ROUND(SUM(Sales), 0) AS Total\_Sales

FROM

SalesByDay

GROUP BY

Day\_of\_Week

ORDER BY

Total\_Sales DESC;