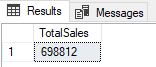
**COFFEE SALES QUERIES**

**KPI’s REQUIREMENT**

1. **Total Sales Analysis**

SELECT ROUND(SUM(Sales),0) AS TotalSales

FROM coffee\_shop\_sales;

****

* Calculate the total sales for each respective month

SELECT

DATEPART(MONTH, transaction\_date) AS Month\_No,

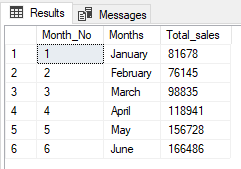
DATENAME(MONTH, transaction\_date) AS Months,

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

FROM coffee\_shop\_sales

GROUP BY DATEPART(MONTH, transaction\_date), DATENAME(MONTH, transaction\_date)

ORDER BY DATEPART(MONTH, transaction\_date);



* Determine the month-to-month increase or decrease in sales + Calculate the difference in sales between the selected month and the previous month.

WITH MonthlySales AS (

SELECT

MONTH(transaction\_date) AS SaleMonth\_No,

DATENAME(MONTH, transaction\_date) AS SalesMonth,

SUM(Sales) AS TotalSales

FROM

coffee\_shop\_sales

GROUP BY

MONTH(transaction\_date), DATENAME(MONTH, transaction\_date)

)

SELECT

SaleMonth\_No,

SalesMonth,

TotalSales,

ISNULL(LAG(TotalSales) OVER (ORDER BY SaleMonth\_No), 0) AS PreviousMonthSales,

ISNULL(TotalSales - LAG(TotalSales) OVER (ORDER BY SaleMonth\_No), 0) AS SalesChange,

CONCAT(ISNULL(ROUND(

CASE

WHEN LAG(TotalSales) OVER (ORDER BY SaleMonth\_No) = 0 THEN 0

ELSE ((TotalSales - LAG(TotalSales) OVER (ORDER BY SaleMonth\_No)) \* 100.0

/ LAG(TotalSales) OVER (ORDER BY SaleMonth\_No))

END

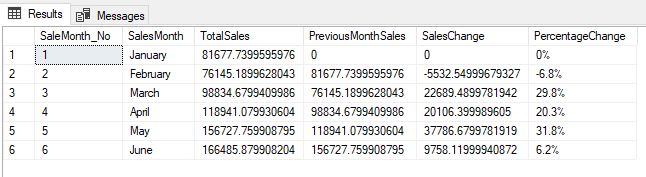
,1) ,0), '%') AS PercentageChange

FROM

MonthlySales

ORDER BY

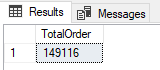
SaleMonth\_No;



1. **Total Order Analysis**

SELECT COUNT(transaction\_id) AS TotalOrder

FROM coffee\_shop\_sales;

****

* Calculate the total number of orders for each respective month

SELECT

MONTH(transaction\_date) AS SaleMonth\_No,

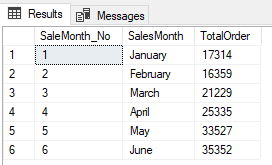
DATENAME(MONTH, transaction\_date) AS SalesMonth,

COUNT(transaction\_id) AS TotalOrder

FROM coffee\_shop\_sales

GROUP BY MONTH(transaction\_date), DATENAME(MONTH, transaction\_date)

ORDER BY SaleMonth\_No;



* Determine the month-to-month increase or decrease in the number of orders + Calculate the difference in the number of orders between the selected month and the previous month.

WITH MonthlyOrders AS

(

SELECT

MONTH(transaction\_date) AS SalesMonth\_No,

DATENAME(MONTH, transaction\_date) AS SalesMonth,

COUNT(transaction\_id) AS TotalOrder

FROM coffee\_shop\_sales

GROUP BY MONTH(transaction\_date), DATENAME(MONTH, transaction\_date)

)

SELECT

SalesMonth\_No,

SalesMonth,

TotalOrder,

ISNULL(LAG(TotalOrder) OVER(ORDER BY SalesMonth\_No, SalesMonth),0) AS PreviousMonthOrder,

ISNULL(TotalOrder - LAG(TotalOrder) OVER(ORDER BY SalesMonth\_No, SalesMonth),0) AS OrderChange,

CONCAT(ISNULL(

CASE

WHEN LAG(TotalOrder) OVER(ORDER BY SalesMonth\_No, SalesMonth) = 0 THEN 0

ELSE ((TotalOrder - LAG(TotalOrder) OVER(ORDER BY SalesMonth\_No, SalesMonth)) \* 100

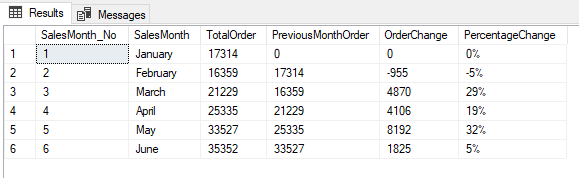
/ LAG(TotalOrder) OVER(ORDER BY SalesMonth\_No, SalesMonth))

END

,0),'%') AS PercentageChange

FROM MonthlyOrders

ORDER BY SalesMonth\_No, SalesMonth;

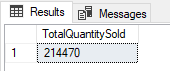


1. **Total Quantity Sold Analysis**

SELECT

SUM(transaction\_qty) AS TotalQuantitySold

FROM coffee\_shop\_sales;

****

* Calculate the total quantity sold for each respective month

SELECT

MONTH(transaction\_date) AS SalesMonth\_No,

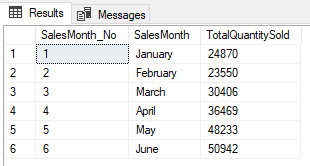
DATENAME(MONTH, transaction\_date) AS SalesMonth,

SUM(transaction\_qty) AS TotalQuantitySold

FROM coffee\_shop\_sales

GROUP BY MONTH(transaction\_date), DATENAME(MONTH, transaction\_date)

ORDER BY SalesMonth\_No;



* Determine the month-to-month increase or decrease in the total quantity sold + Calculate the difference in the total quantity sold between the selected month and the previous month.

SELECT

SalesMonth\_No,

SalesMonth,

TotalQuantitySold,

ISNULL(LAG(TotalQuantitySold) OVER(ORDER BY SalesMonth\_No, SalesMonth),0) AS PrevMonthlyQuantitySold,

ISNULL(TotalQuantitySold - LAG(TotalQuantitySold) OVER(ORDER BY SalesMonth\_No, SalesMonth),0) AS MonthlyQuantitySoldChange,

CONCAT(ISNULL(

CASE

WHEN LAG(TotalQuantitySold) OVER(ORDER BY SalesMonth\_No, SalesMonth) = 0 THEN 0

ELSE (TotalQuantitySold - LAG(TotalQuantitySold) OVER(ORDER BY SalesMonth\_No, SalesMonth)) \* 100

/LAG(TotalQuantitySold) OVER(ORDER BY SalesMonth\_No, SalesMonth)

END

,0),'%')

AS PercentageChange

FROM

(

SELECT

MONTH(transaction\_date) AS SalesMonth\_No,

DATENAME(MONTH, transaction\_date) AS SalesMonth,

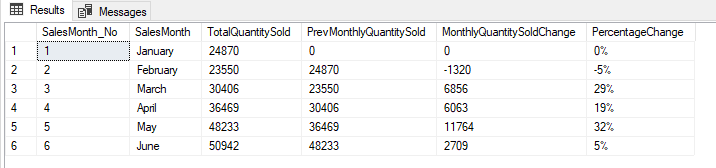
SUM(transaction\_qty) AS TotalQuantitySold

FROM coffee\_shop\_sales

GROUP BY MONTH(transaction\_date), DATENAME(MONTH, transaction\_date)

) AS MonthlyQuantitySold

ORDER BY SalesMonth\_No, SalesMonth;



**CHARTS REQUIREMENT**

1. **Calendar Heat Map**

* Implement a calendar heat map that dynamically adjusts based on the selected month from a slicer.
* Each day on the calendar will be color-coded to represent sales volume, with the darker shades indicating higher sales.
* Implement tooltips to display detailed metrics (Sales, Order and Quantity) when hovering over a specific day.

SELECT

DATEPART(DAY, transaction\_date) AS DayNo,

DATENAME(WEEKDAY, transaction\_date) AS [DayName],

ROUND(SUM(Sales),0) AS TotalSales,

COUNT(transaction\_id) AS TotalOrders,

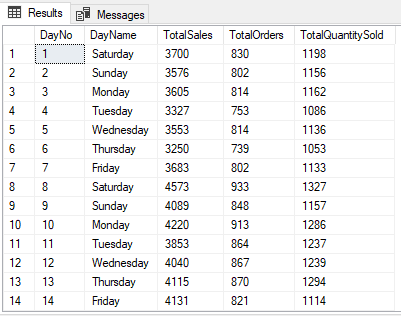
SUM(transaction\_qty) AS TotalQuantitySold

FROM coffee\_shop\_sales

WHERE DATEPART(MONTH, transaction\_date) = 4-----Filter for the Month of April

GROUP BY DATEPART(DAY, transaction\_date), DATENAME(WEEKDAY, transaction\_date)

ORDER BY DayNo;



1. **Sales Analysis by Weekdays and Weekends**

* Segment sales data into weekdays and weekends to analyze performance variations.
* Provide insights into whether sales patterns differ significantly between weekdays and weekends.

SELECT

CASE

WHEN DATEPART(WEEKDAY, transaction\_date) IN (1, 7) THEN 'Weekends' -- Sunday = 1 and Saturday = 7

ELSE 'Weekdays'

END AS DayType,

ROUND(SUM(Sales), 0) AS TotalSales

FROM coffee\_shop\_sales

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

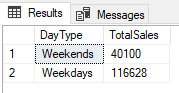
GROUP BY

CASE

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

ELSE 'Weekdays'

END;



1. **Sales Analysis by Store Location**

* Visualize sales data by different store location
* Include month-over-month (MoM) difference metrics based on the selected month in the slicer.
* Highlight MoM sales increase and decrease for each store location to identify trends.

SELECT

store\_location AS StoreLocation,

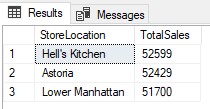
ROUND(SUM(Sales),0) AS TotalSales

FROM coffee\_shop\_sales

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

GROUP BY store\_location

ORDER BY TotalSales DESC;



1. **Daily Sales Analysis with Average Line**

* Display daily sales for the selected month with a line chart.

SELECT

DAY(transaction\_date) AS DayNo,

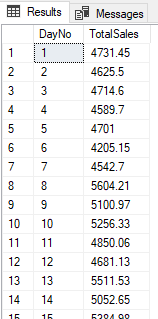
ROUND(SUM(Sales),2) AS TotalSales

FROM coffee\_shop\_sales

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

GROUP BY transaction\_date

ORDER BY DayNo;



* Incorporate an average line on the chart to represent the average daily sales.

**Average Sales**

SELECT

ROUND(AVG(TotalSales)/1000,1) AS AverageSales

FROM

(

SELECT

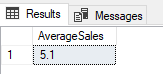
ROUND(SUM(Sales),2) AS TotalSales

FROM coffee\_shop\_sales

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

GROUP BY transaction\_date

) AS Subquery

****

* Highlight bars exceeding or falling below the average sales to identify exceptional sales days.

SELECT

DayNo,

CASE

WHEN TotalSales > AverageSales THEN 'Above Average'

WHEN TotalSales < AverageSales THEN 'Below Average'

ELSE 'Average'

END AS SalesStatus,

TotalSales

FROM

(

SELECT

DAY(transaction\_date) AS DayNo,

ROUND(SUM(Sales),2) AS TotalSales,

AVG(SUM(Sales)) OVER() AS AverageSales

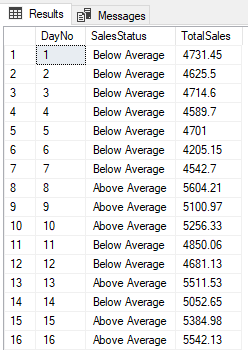
FROM coffee\_shop\_sales

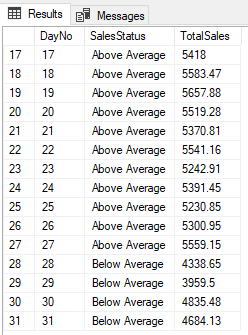
WHERE MONTH(transaction\_date) = 5 ---- For the Month of May

GROUP BY DAY(transaction\_date)

) AS AverageSales

ORDER BY DayNo

****

****

1. **Sales Analysis by Product Category**

* Analyze sales performance across different product categories.
* Provide insights into which product categories contribute the most to overall sales.

SELECT

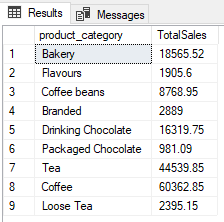
product\_category,

ROUND(SUM(Sales),2) AS TotalSales

FROM coffee\_shop\_sales

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

GROUP BY product\_category



1. **Top 10 products by Sales**

* Identify and display the top 10 products based on sales volume.
* Allow users to quickly visualize the best performing products in terms of sales.

SELECT

TOP (10)

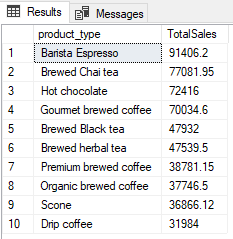
product\_type,

ROUND(SUM(Sales),2) AS TotalSales

FROM coffee\_shop\_sales

GROUP BY product\_type

ORDER BY TotalSales DESC;



1. **Sales Analysis by Days and Hours**

* Utilize a heat map to visualize sales patterns by days and hours.
* Implement tooltips to display detailed metrics (Sales, orders and Quantity) when hovering over a specific day-hour

SELECT

DATENAME(WEEKDAY, transaction\_date) AS [Days],

DATEPART(HOUR,transaction\_time) AS [Hour],

ROUND(SUM(Sales),2) AS TotalSales,

COUNT(\*) AS TotalOrders,

SUM(transaction\_qty) AS TotalQuantitySold

FROM coffee\_shop\_sales

WHERE MONTH(transaction\_date) = 5 AND DATENAME(WEEKDAY, transaction\_date) = 'Monday'

GROUP BY DATENAME(WEEKDAY, transaction\_date), DATEPART(HOUR,transaction\_time)

ORDER BY [Days]

