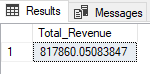
**Pizza Sales SQL queries**

**A. KPI’S:**

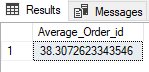
1. *Total Revenue: The sum of total price of pizzas orders.*

SELECT SUM(total\_price) AS Total\_Revenue FROM pizza\_sales;



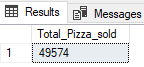
*2. Average Order Value: The average amount spent per order, calculated by dividing the total number of orders.*

SELECT SUM(total\_price) / COUNT(DISTINCT order\_id) AS Average\_Order\_Values FROM pizza\_sales;



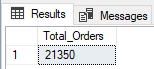
*3. Total Pizzas Sold: The total sum of quantities of all pizzas sold.*

SELECT SUM(quantity) AS Total\_Pizza\_sold FROM pizza\_sales



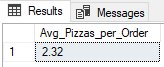
*4. Total Orders Placed: The count of all the orders placed by the customers.*

SELECT COUNT(DISTINCT order\_id) AS Total\_Orders FROM pizza\_sales



*5. Average Pizzas per Order: The average numbers of pizzas sold per order, calculated by dividing the total numbers of pizzas sold by the total number of orders.*

SELECT CAST(CAST(SUM(quantity) AS DECIMAL (10,2))/CAST(COUNT(DISTINCT order\_id) AS DECIMAL (10,2))AS DECIMAL (10,2)) AS Avg\_Pizzas\_per\_Order FROM pizza\_sales



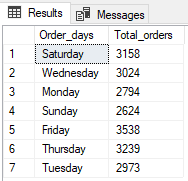
**B. Daily trend for Total orders:**

*1. Create a bar chart that displays the daily trend of total orders over a specific time period. This chart will help us to identify any patterns or fluctuations in the order volume on a daily basis.*

SELECT DATENAME(DW, order\_date) as Order\_days, COUNT(DISTINCT order\_id) AS Total\_orders

FROM pizza\_sales

GROUP BY DATENAME(DW, order\_date)



*\*\* Total orders arranged with respect to Days from Sunday to Saturday*

SELECT DATENAME(DW, order\_date) as Order\_days, COUNT(DISTINCT order\_id) as Total\_orders

FROM pizza\_sales

GROUP BY DATENAME(DW, order\_date)

ORDER BY CASE WHEN DATENAME(DW, order\_date) = 'Sunday' then 1

WHEN DATENAME(DW, order\_date) = 'Monday' then 2

WHEN DATENAME(DW, order\_date) = 'Tuesday' then 3

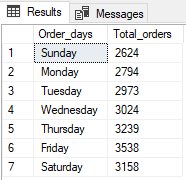
WHEN DATENAME(DW, order\_date) = 'Wednesday' then 4

WHEN DATENAME(DW, order\_date) = 'Thursday' then 5

WHEN DATENAME(DW, order\_date) = 'Friday' then 6

WHEN DATENAME(DW, order\_date) = 'Saturday' then 7

ELSE NULL END

**

**C. Monthly Trend for Orders:**

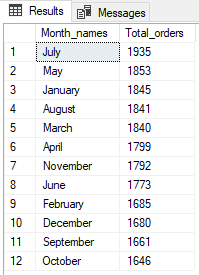
*1. Create a line chart that illustrates the hourly trend of total orders throughout the day. The chart will allow us to identify peak hours or periods of high order activity.*

SELECT DATENAME(MONTH, order\_date) as Month\_names, COUNT(DISTINCT order\_id) as Total\_orders

FROM pizza\_sales

GROUP BY DATENAME(MONTH, order\_date)

ORDER BY Total\_orders DESC



**D. Percentage of Sales by Pizza categories:**

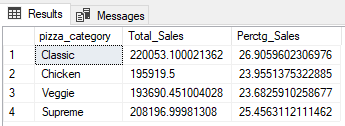
*1. Create a pie chart that shows the distribution of sales across different pizza categories. This chart will provide insights into the popularity of various pizza categories and their contributions to overall sales.*

SELECT pizza\_category, SUM(total\_price) AS Total\_Sales, SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales)

AS Perctg\_Sales

FROM pizza\_sales

GROUP BY pizza\_category



*\*\* In order to filter this data according to each month:*

SELECT pizza\_category, CAST (SUM(total\_price) AS DECIMAL (10,2)) AS Total\_Sales, CAST (SUM(total\_price) \* 100 /

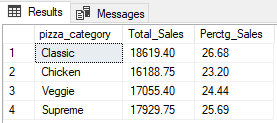
(SELECT SUM(total\_price) FROM pizza\_sales WHERE MONTH(order\_date) = 1) AS DECIMAL (10,2))

AS Perctg\_Sales

FROM pizza\_sales

WHERE MONTH(order\_date) = 1

GROUP BY pizza\_category



**E. Percentage of Sales by Pizza sizes:**

*1. Generate a pie chart that represents the percentage of sales attributed to different pizza sizes. This chart will us to understand customer’s preferences for pizza sizes and their impact on sales.*

SELECT pizza\_size, CAST(SUM(total\_price) AS DECIMAL (10,2)) AS Total\_Revenue,

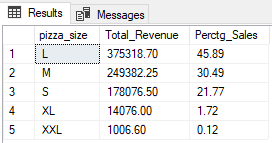
CAST (SUM (total\_price) \* 100 /

(SELECT SUM (total\_price) FROM pizza\_sales)AS DECIMAL (10,2)) AS Perctg\_Sales

FROM pizza\_sales

GROUP BY pizza\_size

ORDER BY pizza\_size

**

**F. Total Pizza Sold by Pizza Catagory:**

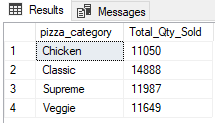
*1. Create a funnel chart that represents the total number of pizza sold for each pizza category. This chart will allow us to compare the sales performance of different pizza categories.*

SELECT pizza\_category, SUM (quantity) AS Total\_Qty\_Sold

FROM pizza\_sales

GROUP BY pizza\_category

ORDER BY pizza\_category



**G. Top 5 best seller by Revenue, Total quantity and Total orders:**

*1. Create a bar chart highlighting the top 5 bestselling pizzas based on the revenue, total quantity, total orders. This chart will help us to identify the most popular pizza options.*

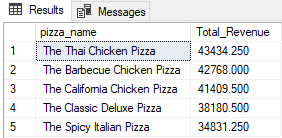
*(i) Top 5 category of pizzas w.r.t Revenue*

SELECT TOP 5 pizza\_name, CAST(SUM(total\_price) AS DECIMAL(10,3)) AS Total\_Revenue

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue DESC

**

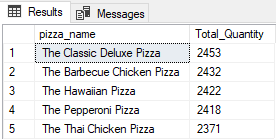
*(ii) Top 5 category of pizzas w.r.t Quantity*

SELECT TOP 5 pizza\_name, CAST(SUM(quantity) AS DECIMAL(10,3)) AS Total\_Quantity

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity DESC



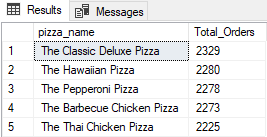
*(iii) Top 5 category of pizzas w.r.t Orders*

SELECT TOP 5 pizza\_name, COUNT(DISTINCT order\_id) AS Total\_Orders

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Orders DESC

**

**H. Bottom 5 best seller by Revenue, Total quantity and Total orders:**

*1. Create a bar chart highlighting the top 5 bestselling pizzas based on the revenue, total quantity, total orders. This chart will help us to identify the most popular pizza options.*

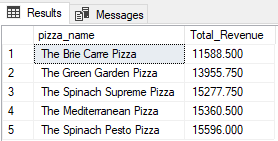
*(i) Top 5 category of pizzas w.r.t Revenue*

SELECT TOP 5 pizza\_name, CAST(SUM(total\_price) AS DECIMAL(10,3)) AS Total\_Revenue

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue ASC

**

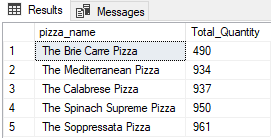
*(ii) Top 5 category of pizzas w.r.t Quantity*

SELECT TOP 5 pizza\_name, CAST(SUM(quantity) AS DECIMAL(10,3)) AS Total\_Quantity

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity ASC



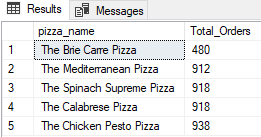
*(iii) Top 5 category of pizzas w.r.t Orders*

SELECT TOP 5 pizza\_name, COUNT(DISTINCT order\_id) AS Total\_Orders

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Orders

**