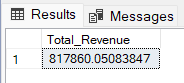
**PIZZA SALES SQL QUERIES**

**A. KPI’s**

**1. Total Revenue:**

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

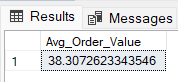
****

**2. Average Order Value**

SELECT SUM(total\_price) / MAX(order\_id) AS Avg\_Order\_Value FROM pizza\_sales;

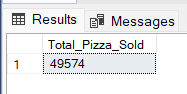
Or

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

****

**3. Total Pizzas Sold**

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

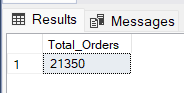
****

**4. Total Orders**

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

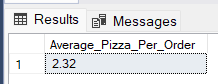
Or

SELECT MAX(order\_id) AS Total\_Orders FROM pizza\_sales;

****

**5. Average Pizzas Per Order**

SELECT CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) / CAST(COUNT(DISTINCT order\_id) AS DECIMAL(10,2)) AS DECIMAL(10,2)) AS Average\_Pizza\_Per\_Order FROM pizza\_sales;

****

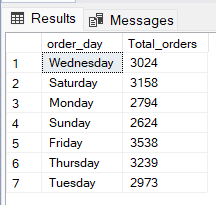
**B. CHARTS REQUIREMENT**

**1. Daily Trend for Total Orders**

SELECT DATENAME(DW, order\_date) AS order\_day, COUNT(DISTINCT order\_id) AS Total\_orders

FROM pizza\_sales

GROUP BY DATENAME(DW, order\_date)

****

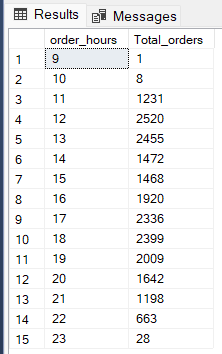
**2. Hourly Trend for Total Orders**

SELECT DATEPART(HOUR, order\_time) AS order\_hours, COUNT(DISTINCT order\_id) AS Total\_orders

FROM pizza\_sales

GROUP BY DATEPART(HOUR, order\_time)

ORDER BY DATEPART(HOUR, order\_time)

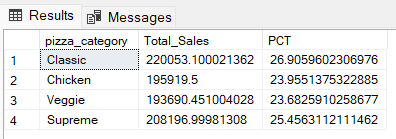
****

**3. Percentage of Sales by Pizza Category**

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

FROM pizza\_sales

GROUP BY pizza\_category

****

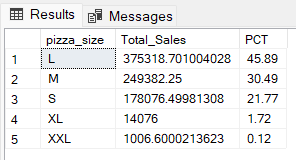
**4. Percentage of Sales by Pizza Size**

SELECT pizza\_size, SUM(total\_price) AS Total\_Sales, CAST(SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales) AS DECIMAL(10,2)) AS PCT

FROM pizza\_sales

GROUP BY pizza\_size

ORDER BY PCT DESC

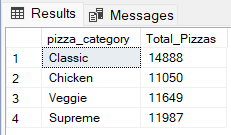
****

**5. Total Pizzas Sold by Pizza Category**

SELECT pizza\_category, SUM(quantity) AS Total\_Pizzas

FROM pizza\_sales

GROUP BY pizza\_category

****

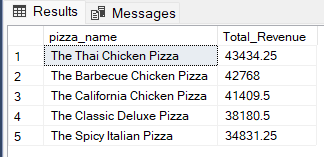
**6. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders**

*A) By revenue*

SELECT TOP 5 pizza\_name, SUM(total\_price) AS Total\_Revenue FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue DESC

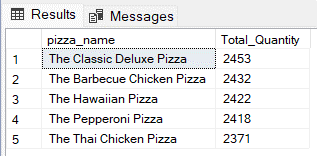
****

*B) By Total Quantity*

SELECT TOP 5 pizza\_name, SUM(quantity) AS Total\_Quantity FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity DESC

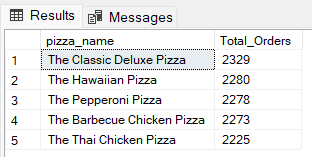


*C) By Total Orders*

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

GROUP BY pizza\_name

ORDER BY Total\_Orders DESC

****

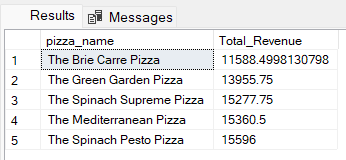
**7. Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders**

*A) By revenue*

SELECT TOP 5 pizza\_name, SUM(total\_price) AS Total\_Revenue FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue

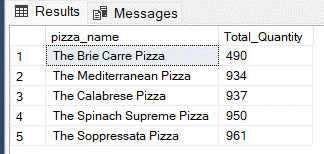
****

*B) By Total Quantity*

SELECT TOP 5 pizza\_name, SUM(quantity) AS Total\_Quantity FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity

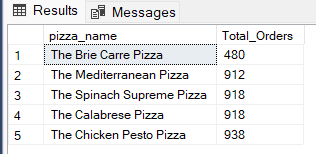


*C) By Total Orders*

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

GROUP BY pizza\_name

ORDER BY Total\_Orders

****

***NOTE***

If you want to apply the Month, Quarter, Week filters to the above queries you

can use WHERE clause. Follow some of below examples

SELECT DATENAME(MONTH, order\_date) AS Month\_name, COUNT(DISTINCT order\_id) AS Total\_orders

FROM pizza\_sales

WHERE MONTH(order\_date) = 1

GROUP BY DATENAME(MONTH, order\_date)

*\*Here MONTH(order\_date) = 1 indicates that the output is for the month of January.*

*MONTH(order\_date) = 4 indicates that the output is for Month of April*

SELECT DATENAME(MONTH, order\_date) AS Month\_name, COUNT(DISTINCT order\_id) AS Total\_orders

FROM pizza\_sales

WHERE DATEPART(QUARTER, order\_date) = 1

GROUP BY DATENAME(MONTH, order\_date)

*\*Here DATEPART(QUARTER, order\_date) = 1 indicates that the output is for the Quarter 1.*

*DATEPART(QUARTER, order\_date) = 3 indicates that the output is for Quarter 3*