**Pizza Sales SQL Queries Presentation with Result**

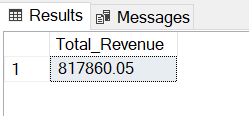
**KPI**

**1.--Total Revenue**

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

FROM pizza\_sales

**Output:**

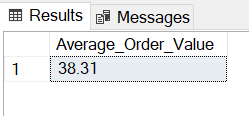


**2. - - Average Order Value**

SELECT SUM(total\_price) / COUNT(DISTINCT order\_id) AS Average\_Order\_Value

FROM pizza\_sales

**Output:**

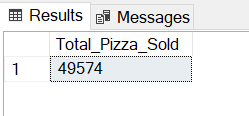


**3.-- Total Pizza Sold**

SELECT SUM(quantity) AS Total\_Pizza\_Sold

FROM pizza\_sales

**Output:**

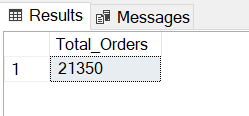


**4. --Total Orders**

SELECT count(distinct order\_id) AS Total\_Orders

FROM pizza\_sales

**Output:**



**5.--Average Pizza Per Orders**

--Note: This result shows exact number but not in fraction

--That's why we have to cast this item

--10,2 decimal meaning after comma it will shows 10 digit

--So we need to cast again to 2 decimal point

SELECT CAST( --this is for 2 decimal

CAST( --this is for total output will be decimal

SUM(quantity) AS DECIMAL (10,2)) /

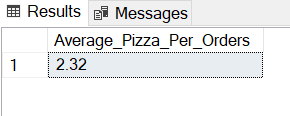
CAST( --this is for total output will be decimal

COUNT (DISTINCT order\_id) AS DECIMAL (10,2))

AS DECIMAL(10,2)) AS Average\_Pizza\_Per\_Orders --this is for 2 decimal

FROM pizza\_sales

**Output:**



**B. Hourly Trend Total Pizza Sold**

--Datepart will extract hour data from order\_time

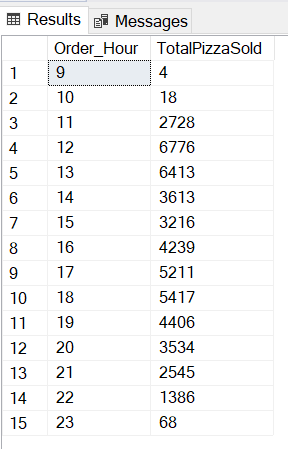
SELECT DATEPART(HOUR, order\_time) AS Order\_Hour, SUM(quantity) AS TotalPizzaSold

FROM pizza\_sales

GROUP BY DATEPART(HOUR, order\_time)

ORDER BY DATEPART(HOUR, order\_time)

**Output:**



**C. Weekly Trend Total Pizza Sold**

--Datepart will extract IS\_Week data from order\_time

SELECT DATEPART(ISO\_WEEK, order\_date) AS Week\_Number, --it will shows total weeks of a year

YEAR(order\_date) as Order\_Year, --years of 2015 showed

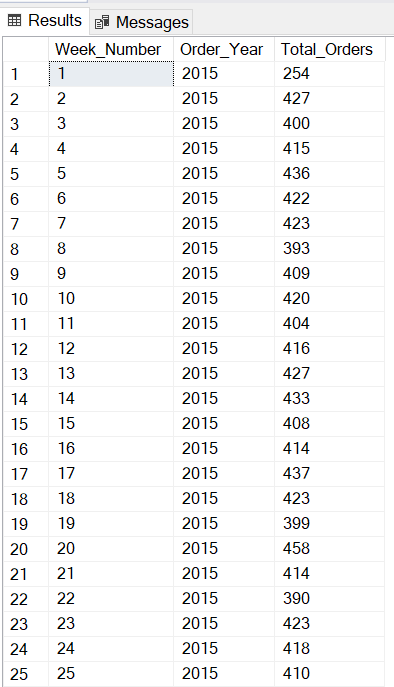
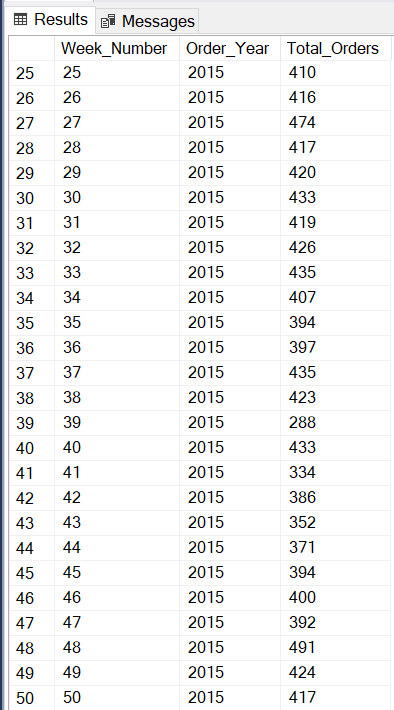
count(distinct order\_id) AS Total\_Orders

FROM pizza\_sales

GROUP BY DATEPART(ISO\_WEEK, order\_date), YEAR(order\_date) --aggregated function are used as a group by statement (requirement)

ORDER BY DATEPART(ISO\_WEEK, order\_date), YEAR(order\_date) --aggregated function are used as a order by statement (requirement)

**Output:**

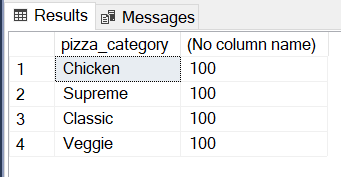
**D.1. Percentage of Total Order by Pizza Category**

SELECT pizza\_category, SUM(total\_price) \* 100 / SUM(total\_price)

FROM pizza\_sales

GROUP BY pizza\_category

**Output:**



--Note: The above result shows only 100 with pizza category

--So we should be subquery to find out real values

--just divided by subquery sum\_of\_total\_Price

SELECT pizza\_category AS P\_Categrory, ROUND(sum(total\_price),2) as Total\_Sales , ROUND( SUM(total\_price) \* 100 /

-- A subquery used here for correct result

(SELECT SUM(total\_price) from pizza\_sales WHERE MONTH(order\_date) = 1 ),2) AS Percentage\_OfTotalSales

FROM pizza\_sales

--when we used filter in main query we should use this filter also subqueyr

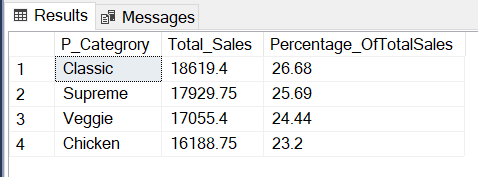
WHERE MONTH(order\_date) = 1

--WHERE MONTH(order\_date) = 1 indicateds that will shows the month of january = 4 shows the month of April

GROUP BY pizza\_category

ORDER BY Percentage\_OfTotalSales DESC, Total\_Sales DESC

**Output:**



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

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

SUM(total\_price) \* 100 /

-- A subquery used here for correct result

-- Both subquery and main query should use filter (SELECT SUM(total\_price) FROM pizza\_sales WHERE DATEPART(QUARTER, order\_date) = 1) AS decimal (10,2) ) AS Percent\_SalesBYPizza\_Size --rounded query end this line

FROM pizza\_sales

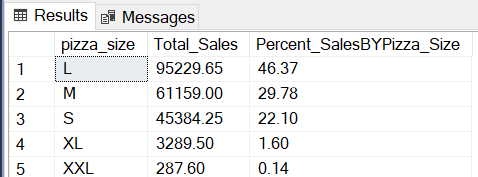
--Where quarter is 1

WHERE DATEPART(QUARTER, order\_date) = 1

GROUP BY pizza\_size

ORDER BY pizza\_size, Total\_Sales DESC

**Output:**



**F.1. Top 5 Best Seller Pizza By Revenue**

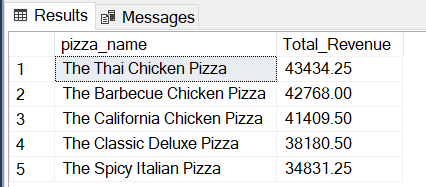
SELECT TOP 5 pizza\_name, CAST(sum(total\_price) AS DECIMAL (10,2)) AS Total\_Revenue

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue DESC

**Output:**



**F.2. Bottom 5 Best Seller Pizza by Revenue**

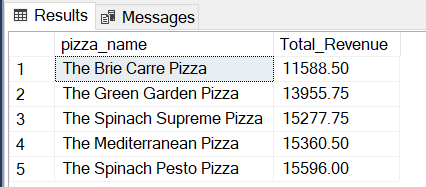
SELECT TOP 5 pizza\_name, CAST(sum(total\_price) AS DECIMAL (10,2)) AS Total\_Revenue

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Revenue ASC --to figur out bottom 5 pizza

**Output:**



**F.3. Top 5 Best Seller Pizza By Total Quantity**

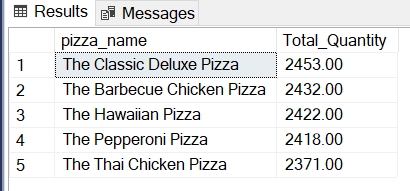
SELECT TOP 5 pizza\_name, CAST(sum(quantity) AS DECIMAL (10,2)) AS Total\_Quantity

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity DESC

**Output:**



**F.5. Bottom 5 Best Seller Pizza By Total Quantity**

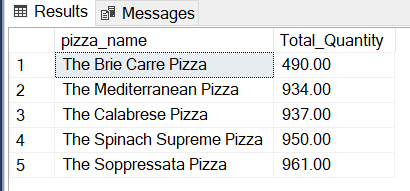
SELECT TOP 5 pizza\_name, CAST(sum(quantity) AS DECIMAL (10,2)) AS Total\_Quantity

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Quantity ASC --to figur out bottom 5 pizza

**Output:**



**F.6. Top 5 Best Seller Pizza By Total Quantity**

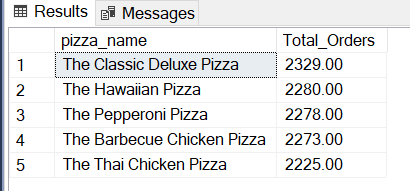
SELECT TOP 5 pizza\_name, CAST(count(distinct order\_id) AS DECIMAL (10,2)) AS Total\_Orders

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Orders DESC

**Output:**



**F.7. Bottom 5 Best Seller Pizza By Total Quantity**

SELECT TOP 5 pizza\_name, CAST(count(distinct order\_id) AS DECIMAL (10,2)) AS Total\_Orders

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Orders ASC --to figur out bottom 5 pizza

**Output:**

