**KPI**

**1. Revenue Growth Rate**

* Formula: ((Current Period Revenue - Previous Period Revenue) / Previous Period Revenue) \* 100
* Explanation: Measures the increase or decrease in revenue over a specific period, indicating overall sales growth.

**2. Customer Retention Rate**

* Formula: (Number of Returning Customers / Total Unique Customers) \* 100
* Explanation: Shows the percentage of customers who make repeat purchases, reflecting customer loyalty.

**3. Average Order Value (AOV)**

* Formula: Total Revenue / Total Number of Orders
* Explanation: Tracks the average revenue per order, highlighting customer spending patterns.

**4. Average Pizzas per Order**

* Formula: Total Pizzas Sold / Total Orders
* Explanation: Indicates the average number of pizzas ordered per transaction, useful for inventory planning.

**5. Order Frequency Rate**

* Formula: Total Orders / Total Unique Customers
* Explanation: Measures the average number of orders placed by each customer, helpful for understanding purchasing habits.

**6. Sales by Pizza Category Percentage**

* Formula: (Revenue from Each Category / Total Revenue) \* 100
* Explanation: Shows the share of revenue each pizza category contributes, helping identify popular products.

**7. Peak Order Hours**

* Formula: Highest Count of Orders by Hour
* Explanation: Identifies the time periods with the highest order frequency, useful for staffing and marketing efforts.

**8. Monthly Sales Growth Rate**

* Formula: ((Current Month Sales - Previous Month Sales) / Previous Month Sales) \* 100
* Explanation: Tracks month-over-month growth in sales, showing seasonal or monthly demand trends.

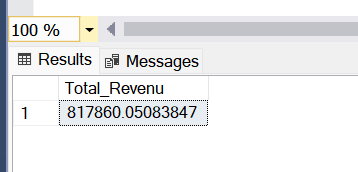
**9. Top-Selling Pizza Variety**

* Formula: Pizza with Highest Quantity Sold
* Explanation: Identifies the most popular pizza by quantity sold, useful for promotions and menu decisions.

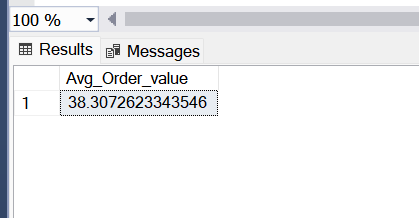
**10. Bottom-Selling Pizza Variety**

* Formula: Pizza with Lowest Quantity Sold
* Explanation: Identifies the least popular pizza, which may need adjustments or promotions.

[here is the dataset PIZZA sale docment!](https://docs.google.com/spreadsheets/d/e/2PACX-1vSuXkC3cOLDucfN7jL9rCE-JRjJ7UbbOHMR4nUrCV23y2Gsi-moY2CniC8ah6lcSw/pubhtml)  
[A Video of how we did!](https://www.linkedin.com/posts/dharani-pulipaka-46ab57264_datavisualization-exceldashboard-sql-activity-7258217493033291776-rilC?utm_source=share&utm_medium=member_desktop)  
  
  
--SELECT \* FROM pizza\_sales  
 **1-Total revenue**

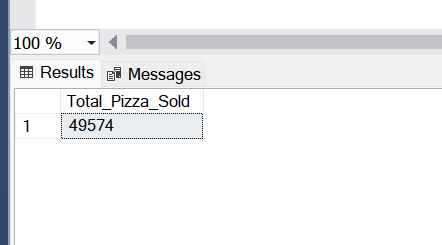
SELECT SUM(total\_price) AS Total\_Revenu from pizza\_sales  
***output***   
**Explanation:**

* This query calculates the total revenue by summing up all values in the total\_price column of the pizza\_sales table.
* SUM(total\_price) adds up the cost of each sale.
* We use AS Total\_Revenue to label the output as Total\_Revenue. **2- Average order value**

SELECT SUM(total\_price) / COUNT(DISTINCT order\_id) AS Avg\_Order\_value FROM pizza\_sales  
***Output***  
  


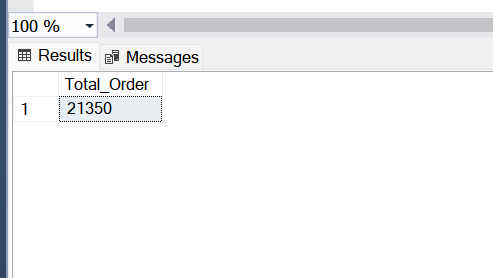
**Explanation:**

* Here, we calculate the average value per order.
* SUM(total\_price) gives the total revenue, while COUNT(DISTINCT order\_id) counts unique orders.
* Dividing these two gives the average revenue per order.
* AS Avg\_Order\_value names this calculated value Avg\_Order\_value. **3- Total PIZZAS sold**  
    
  SELECT SUM(quantity) As Total\_Pizza\_Sold FROM pizza\_sales  
  ***Output***



**Explanation:**

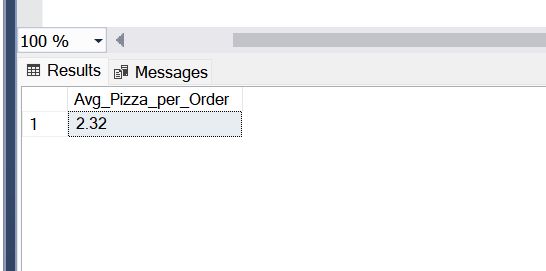
* This query gives the total number of pizzas sold by summing up all values in the quantity column.
* SUM(quantity) totals the pizza quantities across all sales records.
* AS Total\_Pizza\_Sold labels the output as Total\_Pizza\_Sold.

**4-Total Orders**SELECT COUNT(DISTINCT order\_id) AS Total\_Order FROM pizza\_sales  
  
**Output  
  
  
Explanation:**

* Here, we find the total number of unique orders.
* COUNT(DISTINCT order\_id) counts the unique order IDs, showing the total orders placed.
* AS Total\_Order names this count as Total\_Order.

**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 Avg\_Pizza\_per\_Order FROM pizza\_sales  
 **Output**

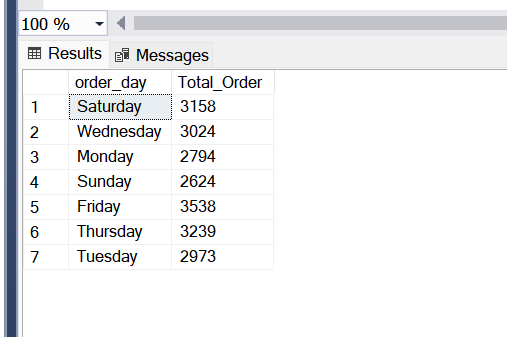


**Explanation:**

* This query calculates the average number of pizzas per order.
* SUM (quantity) gets the total pizzas sold, while COUNT(DISTINCT order\_id) finds the total orders.
* Dividing these gives the average, and CAST(... AS DECIMAL(10,2)) ensures it’s formatted to two decimal places.
* AS Avg\_Pizza\_per\_Order labels the output.

-1-daily trend for total orders  
SELECT DATENAME(DW, order\_date) AS order\_day, COUNT(DISTINCT order\_id) AS Total\_Order

FROM pizza\_sales

GROUP BY DATENAME(DW, order\_date)  
  
  
output  


**Explanation**:

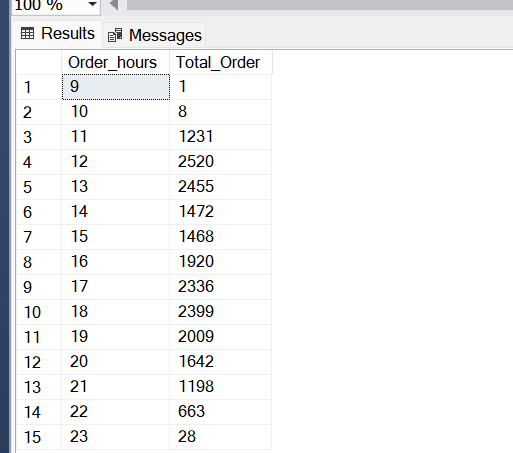
* This query finds the **total orders per day of the week**.
* DATENAME(DW, order\_date) extracts the weekday name (e.g., "Monday") from order\_date.
* COUNT(DISTINCT order\_id) gives the unique orders per day.
* GROUP BY DATENAME(DW, order\_date) groups results by day of the week.

-2-Hourly trend for total orders  
SELECT DATEPART(HOUR, order\_TIME) AS Order\_hours, COUNT(DISTINCT order\_id) AS Total\_Order

FROM pizza\_sales

GROUP BY DATEPART(HOUR, order\_TIME)

ORDER BY DATEPART(HOUR, order\_TIME)

Output  
  
**Explanation**:

* This query finds the **total orders placed each hour**.
* DATEPART(HOUR, order\_TIME) extracts the hour from order\_TIME.
* COUNT(DISTINCT order\_id) counts the unique orders per hour.
* GROUP BY DATEPART(HOUR, order\_TIME) groups results by hour, while ORDER BY DATEPART(HOUR, order\_TIME) sorts them chronologically.

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

SELECT

pizza\_category,

SUM(total\_price) AS Total\_sales,

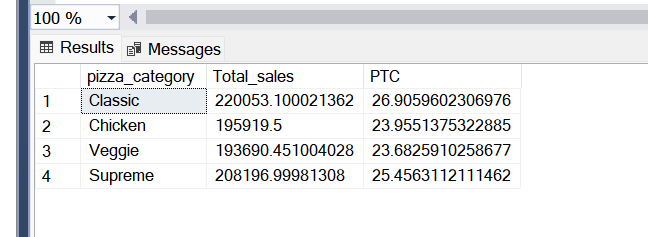
SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales WHERE MONTH(order\_date) = 1) AS PTC

FROM

pizza\_sales

WHERE

MONTH(order\_date) = 1  
GROUP BY pizza\_Category

Output  


**Explanation**:

* This query shows **sales percentage by pizza category for January**.
* SUM(total\_price) gets total sales for each pizza\_category.
* (SUM(total\_price) \* 100 / (SELECT SUM(total\_price) FROM pizza\_sales WHERE MONTH(order\_date) = 1)) calculates each category's percentage of the total January sales.
* GROUP BY pizza\_category groups results by pizza category.

9. **Total Pizzas Sold by Size**   
SELECT

pizza\_size,

SUM(total\_price) AS Total\_sales,

CAST(SUM(total\_price) \* 100.0 / (SELECT SUM(total\_price) FROM pizza\_sales) AS DECIMAL(10,2)) AS PCT

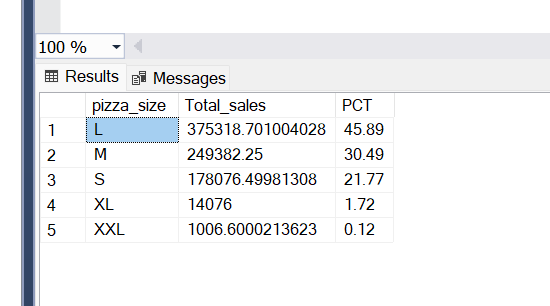
FROM

pizza\_sales

GROUP BY

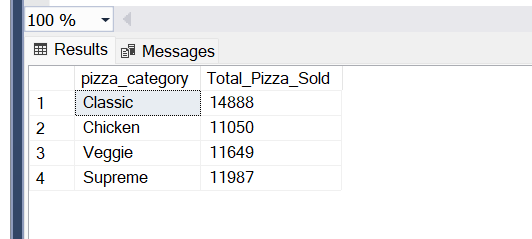
pizza\_size

ORDER BY

PCT DESC;  
-Output  


**Explanation**:

* This query calculates **total sales and percentage for each pizza size**.
* SUM(total\_price) totals sales for each size, while PCT shows each size's share of total sales.
* ORDER BY PCT DESC sorts sizes by percentage in descending order.

10. **Total Pizzas Sold by Category**   
SELECT pizza\_category,

SUM(quantity) as Total\_Pizza\_Sold

FROM pizza\_sales

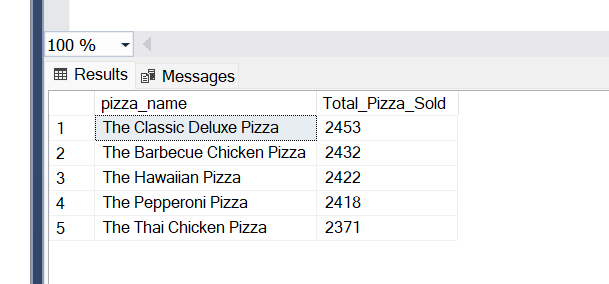
GROUP BY pizza\_category  
output  
  
11. **Top-5 Best Sellers by Total Pizzas Sold**

SELECT TOP 5 pizza\_name,

SUM(quantity) AS Total\_Pizza\_Sold

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Pizza\_Sold DESC  
Output  
  
**Explanation**:

* This query finds the **top 5 best-selling pizzas**.
* SUM(quantity) totals sales for each pizza, and ORDER BY Total\_Pizza\_Sold DESC ranks them from highest to lowest.
* TOP 5 limits results to the five best sellers.

12. **Bottom-5 Pizzas Sold**  
SELECT TOP 5 pizza\_name,

SUM(quantity) AS Total\_Pizza\_Sold

FROM pizza\_sales

GROUP BY pizza\_name

ORDER BY Total\_Pizza\_Sold  
output  


**Explanation**:

* This query finds the **bottom 5 pizzas in terms of sales**.
* SUM(quantity) totals sales for each pizza, and ORDER BY Total\_Pizza\_Sold ranks them from lowest to highest.
* TOP 5 limits results to the five lowest-selling pizzas.