PIZZA SALES ANALYSIS USING SQL

SOFTWARE USED:

MS EXCEL.
MS SQL SERVER.
SQL SERVER MANAGEMENT STUDIO.
POWER BI.

● WHERE EVERY SLICE TELLS A STORY

INTRODUCTION:

This project focuses on analyzing pizza sales data using Structured Query Language (SQL). The primary goal is to gain actionable insights into sales patterns, customer preferences, and overall business performance. The dataset includes detailed records of pizza orders, including order dates, customer information, types of pizzas sold, and revenue generated.

By leveraging SQL queries, the analysis covers key areas such as:

- Identifying the most popular pizza types.
- Examining peak sales periods.
- Analyzing sales performance by size, toppings, and regions.
- Highlighting trends to optimize inventory and improve marketing strategies.

The insights derived from this analysis can help pizza businesses enhance customer satisfaction, streamline operations, and increase profitability.

PROBLEM STATEMENT

KPI’s REQUIREMENT

We need to analyse key indicators for our pizza sales data to gain insights into our business performance. Specifically, we want to calculate the following metrics :

01 TOTAL REVENUE : The sum of total price of all pizza orders.

02 AVERAGE ORDER VALUE : The average amount spent per order, calculated by dividing the total revenue by the total number of orders.

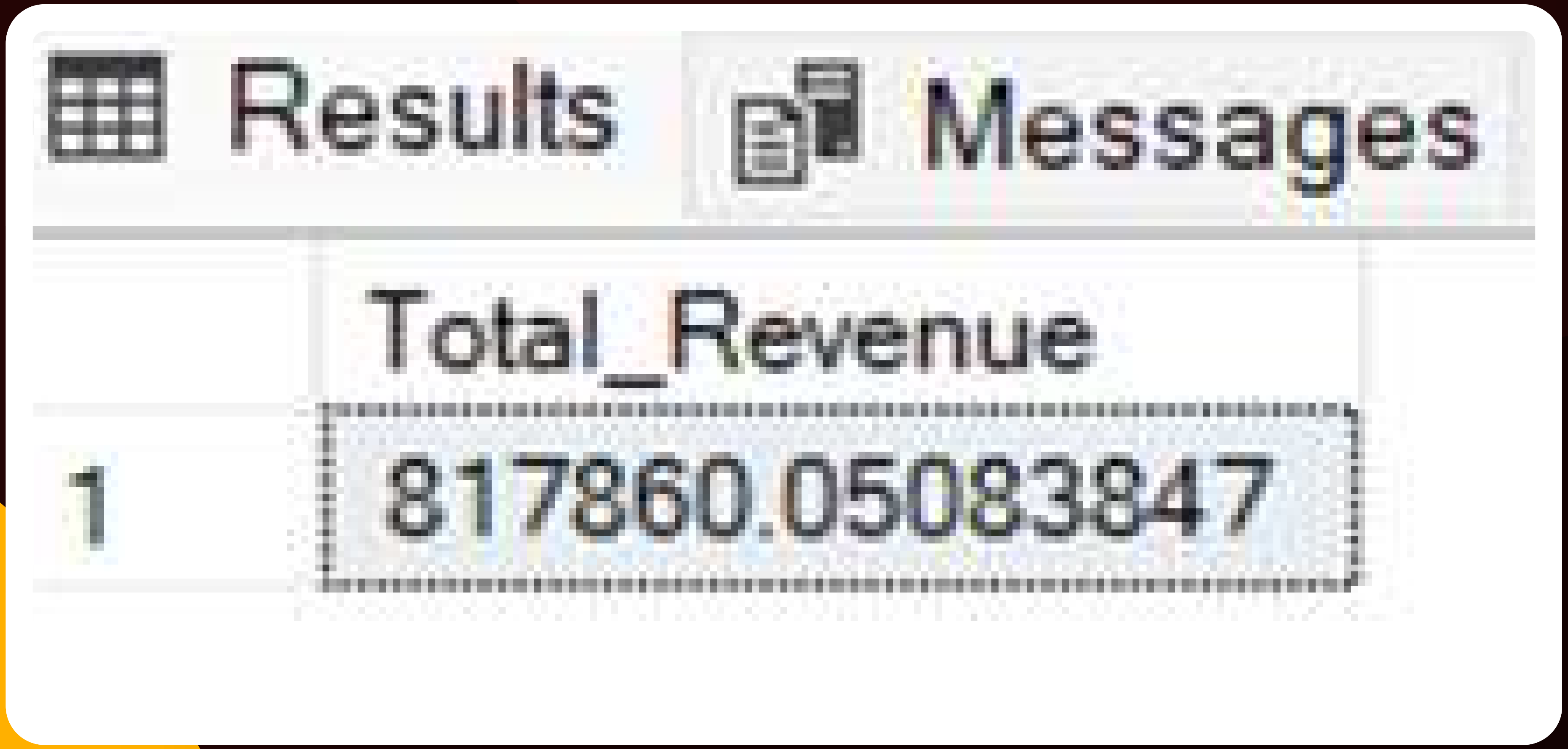
03 TOTAL PIZZA SOLD : The sum of the quantities of all pizzas sold.

04 TOTAL ORDERS : The total number of orders placed.

05 AVERAGE PIZZAS PER ORDER : The average number of pizzas sold per order, calculated by dividing the total number of pizzas sold by the total number of orders.

01. TOTAL REVENUE : THE SUM OF TOTAL PRICE OF ALL PIZZA ORDERS.

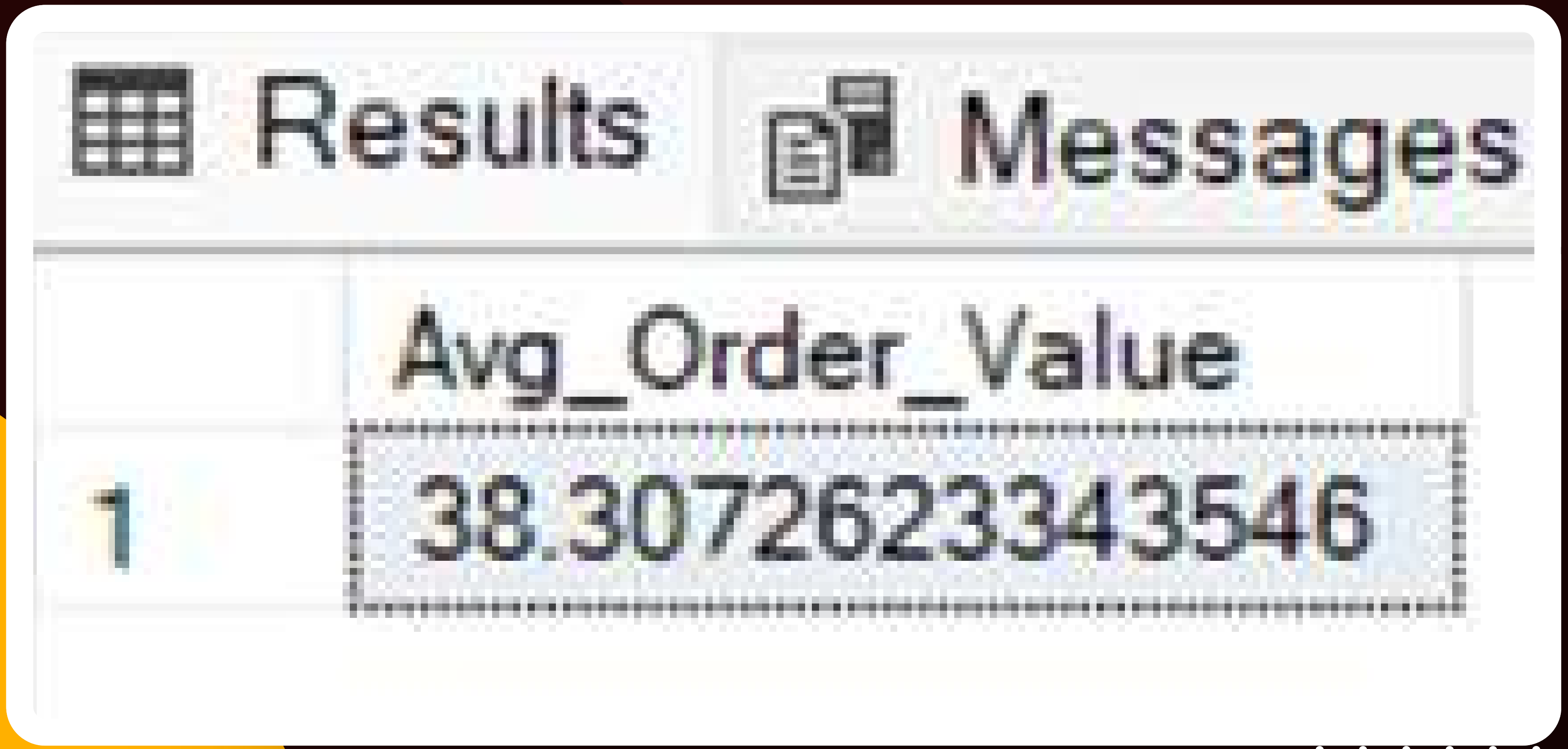
```
SELECT SUM(TOTAL_PRICE) AS
TOTAL_REVENUE FROM PIZZA_SALES
```



	Total_Revenue
1	817860.05083847

02. AVERAGE ORDER VALUE : THE AVERAGE AMOUNT SPENT PER ORDER, CALCULATED BY DIVIDING THE TOTAL REVENUE BY THE TOTAL NUMBER OF ORDERS.

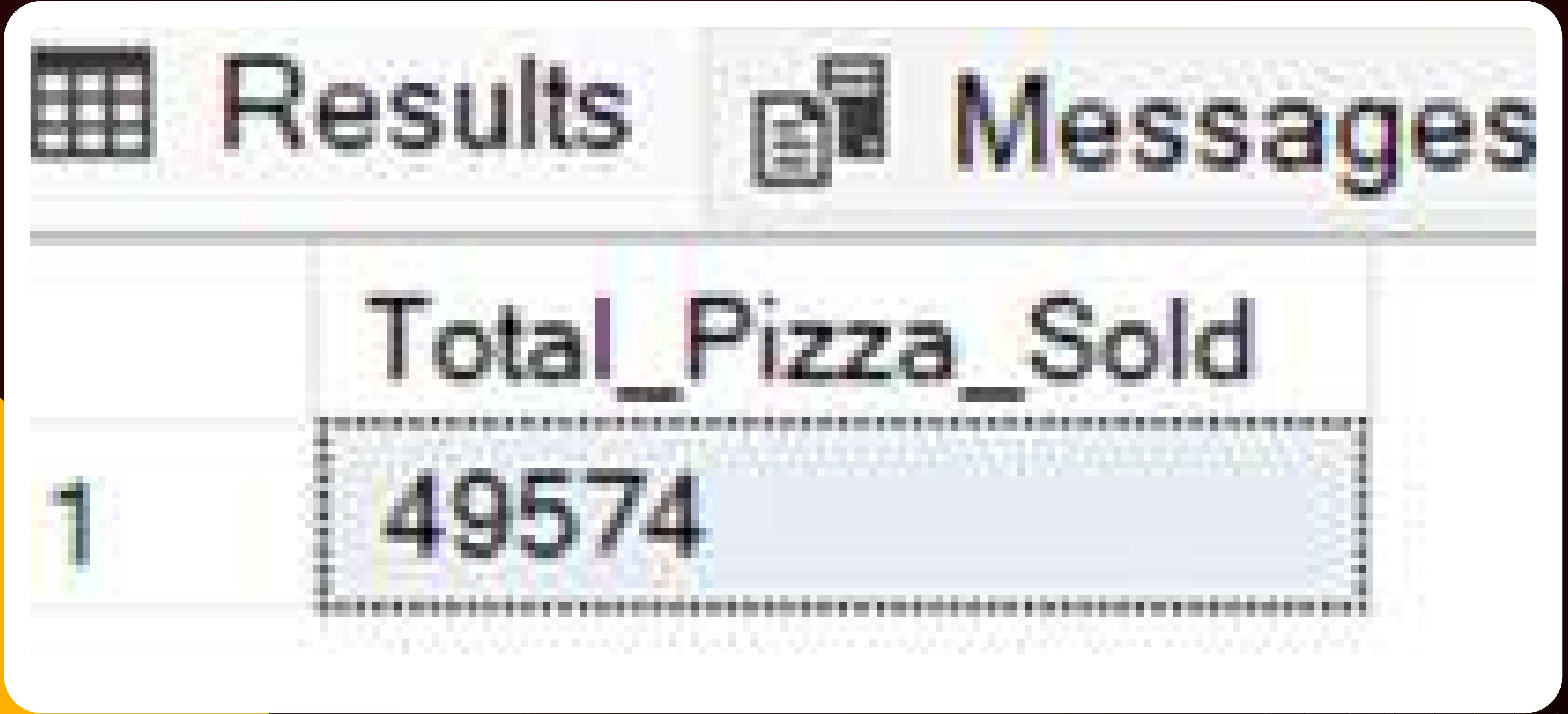
```
SELECT SUM(TOTAL_PRICE) /
COUNT(DISTINCT ORDER_ID) AS
AVG_ORDER_VALUE FROM PIZZA_SALES
```



```
   Avg_Order_Value
1  38.3072623343546
```

03. TOTAL PIZZA SOLD : THE SUM OF THE QUANTITIES OF ALL PIZZAS SOLD.

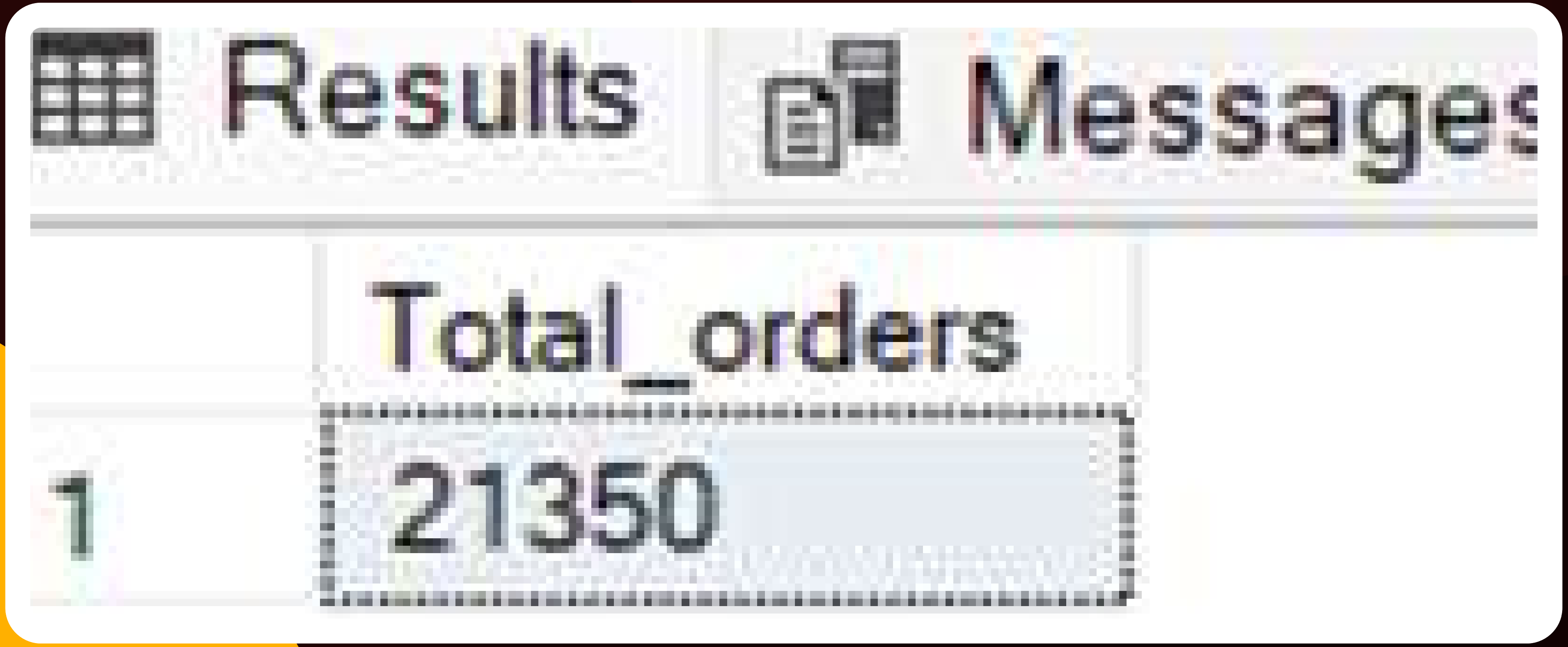
```
SELECT SUM(QUANTITY) AS
TOTAL_PIZZA_SOLD FROM PIZZA_SALES
```



	Total_Pizza_Sold
1	49574

04. TOTAL ORDERS : THE TOTAL NUMBER OF ORDERS PLACED.

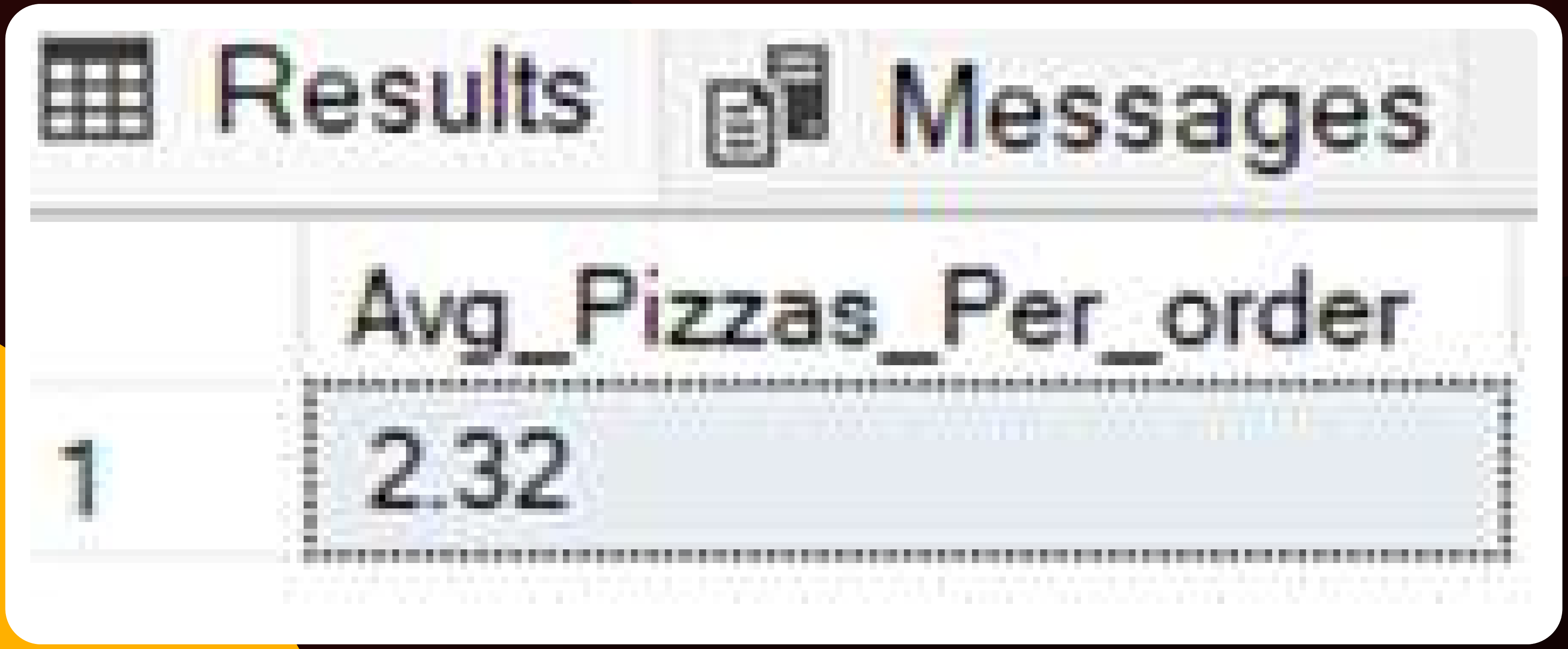
```
SELECT COUNT(DISTINCT ORDER_ID) AS
TOTAL_ORDERS FROM PIZZA_SALES
```



```
  Total_orders
1 21350
```

05. AVERAGE PIZZAS PER ORDER : THE AVERAGE NUMBER OF PIZZAS SOLD PER ORDER, CALCULATED BY DIVIDING THE TOTAL NUMBER 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
```



	Avg_Pizzas_Per_order
1	2.32

PROBLEM STATEMENT

CHARTS REQUIREMENT

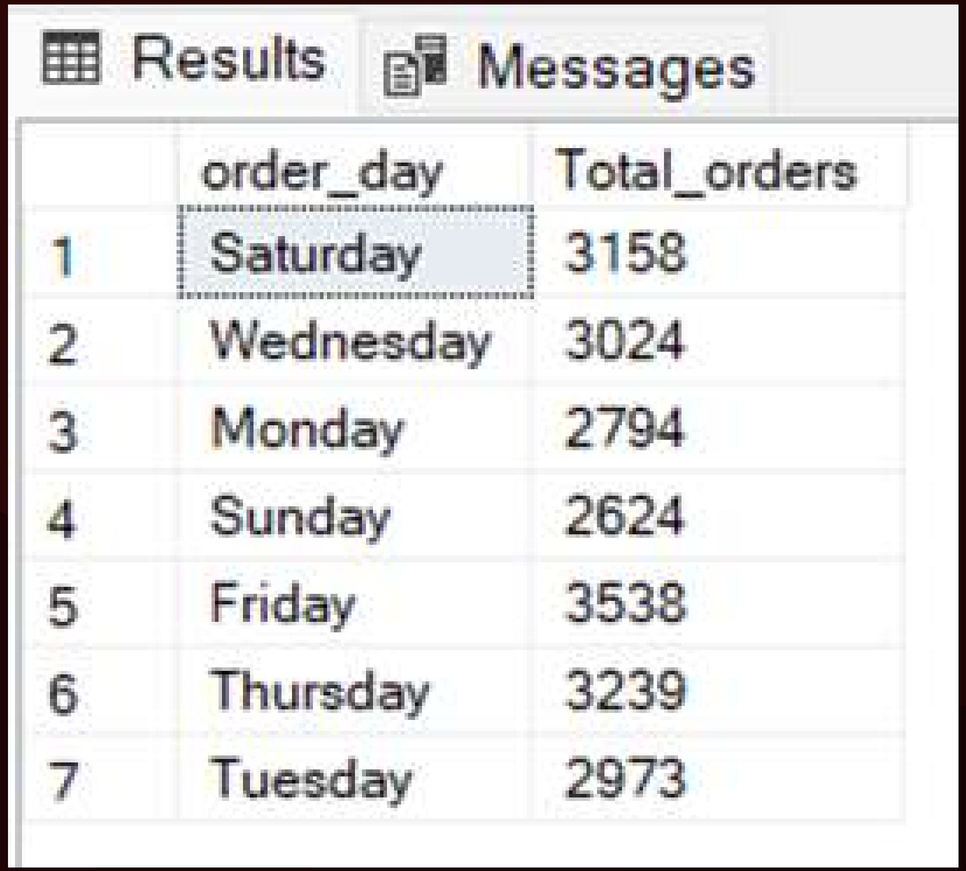
We would like to visualize various aspects of our pizza sales data to gain insights and understand key trends. We have identified the following requirements for creating charts :

01 DAILY TREND FOR TOTAL ORDERS :

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

OUTPUT :

```
SELECT DATENAME(DW, order_date) as
order_day, COUNT(DISTINCT order_id) AS
Total_orders from pizza_sales
GROUP BY DATENAME(DW, order_date)
```



	order_day	Total_orders
1	Saturday	3158
2	Wednesday	3024
3	Monday	2794
4	Sunday	2624
5	Friday	3538
6	Thursday	3239
7	Tuesday	2973

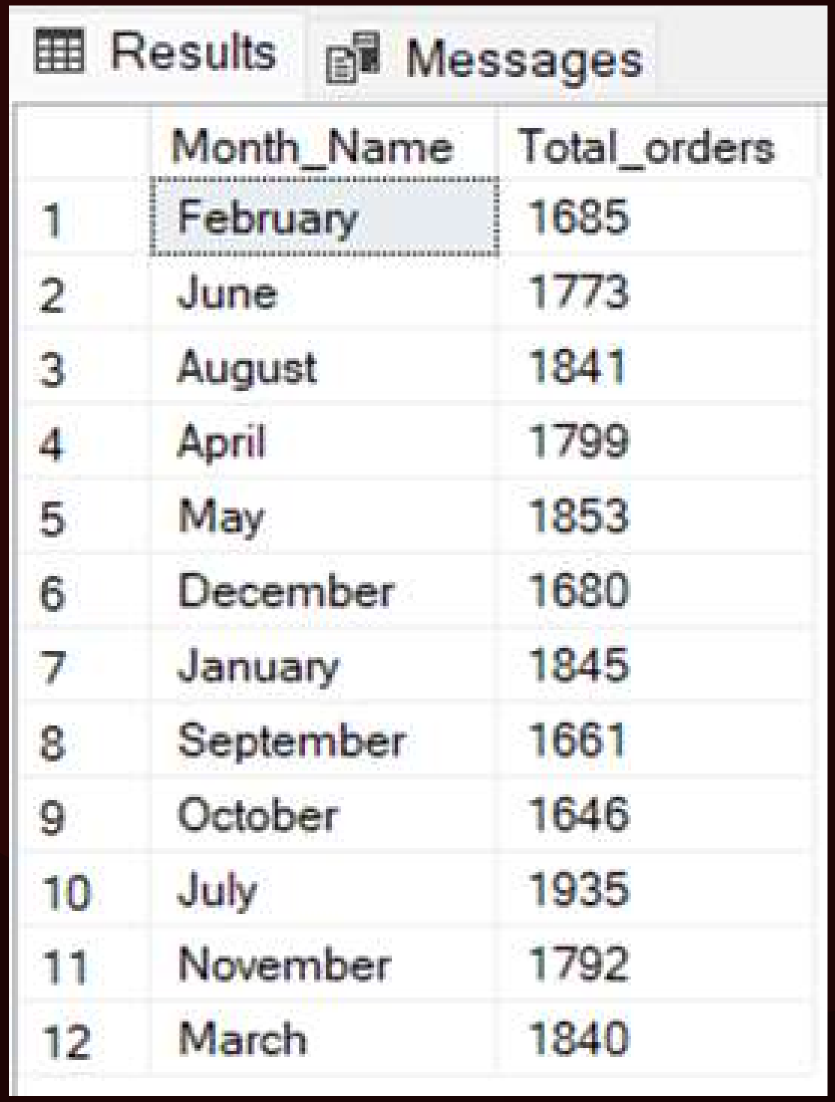
PROBLEM STATEMENT

02 MONTHLY TREND FOR TOTAL ORDERS :

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

OUTPUT :

```
SELECT DATENAME(MONTH, order_date) as
Month_Name, COUNT(DISTINCT order_id)
AS Total_orders from pizza_sales
GROUP BY DATENAME(MONTH, order_date)
```



	Month_Name	Total_orders
1	February	1685
2	June	1773
3	August	1841
4	April	1799
5	May	1853
6	December	1680
7	January	1845
8	September	1661
9	October	1646
10	July	1935
11	November	1792
12	March	1840

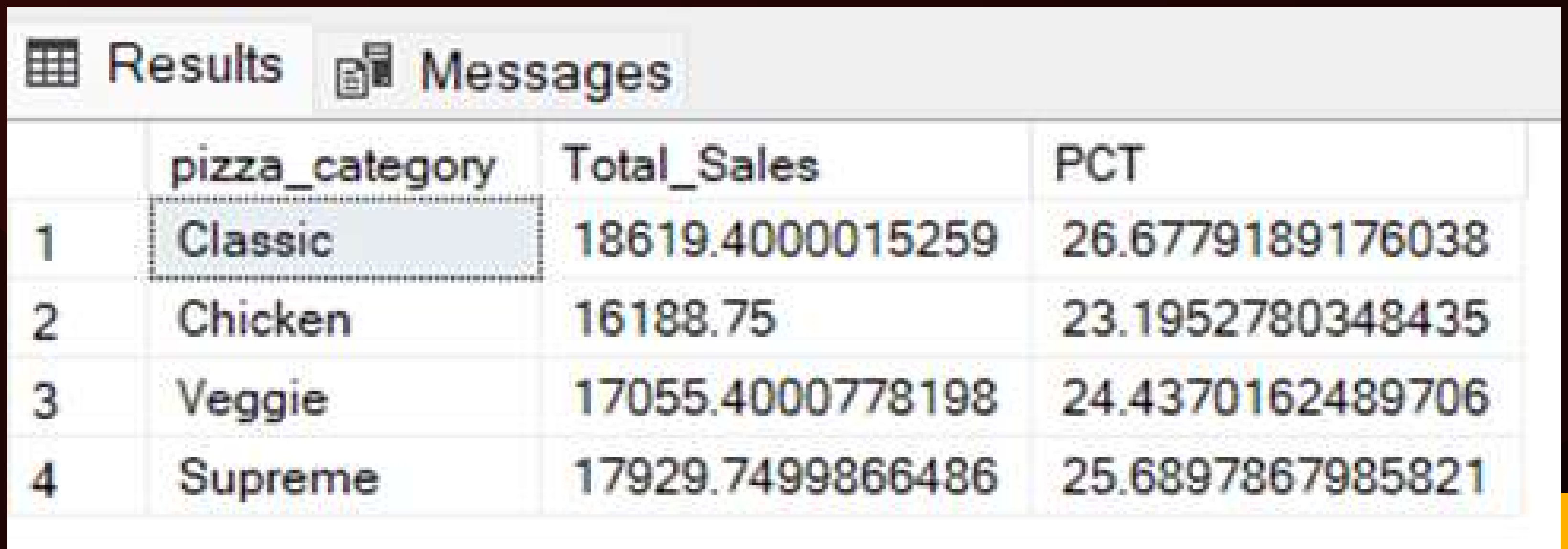
PROBLEM STATEMENT

03 % OF SALES BY PIZZA CATEGORY:

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 contribution to overall sales.

OUTPUT :

```
SELECT pizza_size, CAST(sum(total_price) AS DECIMAL(10,2))
as Total_Sales, CAST(sum(total_price)*100 /
(SELECT sum(total_price) from pizza_sales WHERE
DATEPART(quarter, order_date)=1) AS DECIMAL(10,2)) as PCT
from pizza_sales
WHERE DATEPART(quarter, order_date)=1
GROUP BY pizza_size
ORDER BY PCT DESC
```



	pizza_category	Total_Sales	PCT
1	Classic	18619.4000015259	26.6779189176038
2	Chicken	16188.75	23.1952780348435
3	Veggie	17055.4000778198	24.4370162489706
4	Supreme	17929.7499866486	25.6897867985821

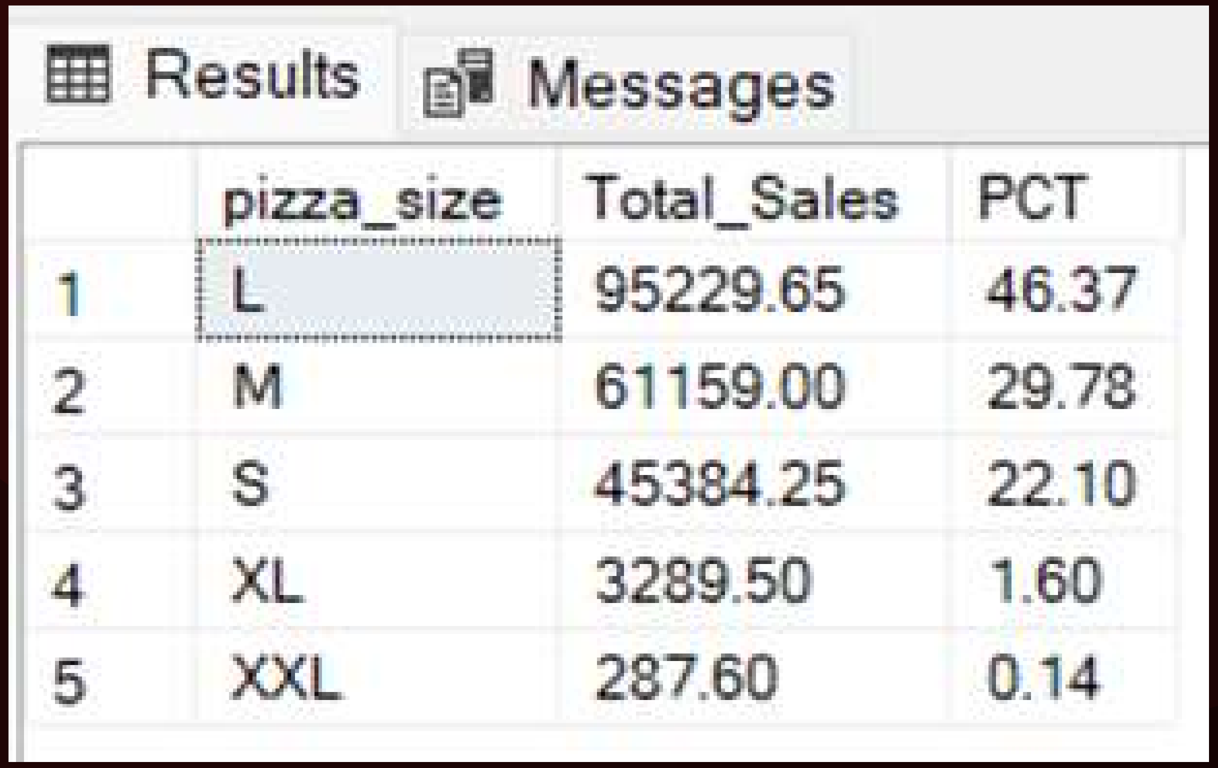
PROBLEM STATEMENT

04 % OF SALES BY PIZZA SALES:

Generate a ple chart that represents the percentage of sales attributed to different pizza sizes. This chart will help us understand customer preferences for pizza sizes and their impact on sales.

OUTPUT :

```
SELECT pizza_size, CAST(sum(total_price) AS DECIMAL(10,2))
as Total_Sales, CAST(sum(total_price)*100 /
(SELECT sum(total_price) from pizza_sales WHERE
DATEPART(quarter, order_date)=1) AS DECIMAL(10,2)) as PCT
from pizza_sales
WHERE DATEPART(quarter, order_date)=1
GROUP BY pizza_size
ORDER BY PCT DESC
```



	pizza_size	Total_Sales	PCT
1	L	95229.65	46.37
2	M	61159.00	29.78
3	S	45384.25	22.10
4	XL	3289.50	1.60
5	XXL	287.60	0.14

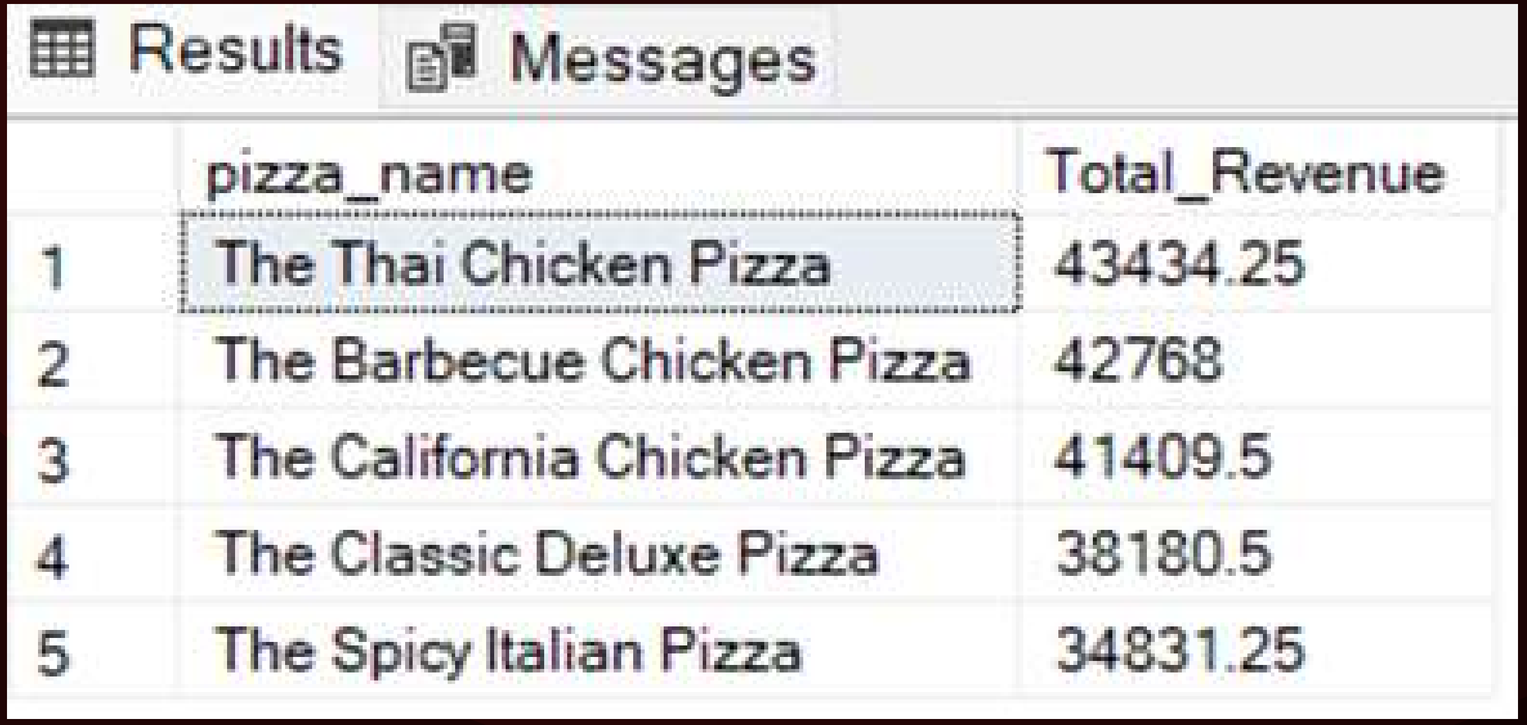
PROBLEM STATEMENT

05 TOP 5 BEST SELLER BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS:

Create a bar chart highlighting the top 5 best-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will help us identify the most popular pizza options.

OUTPUT :

```
SELECT TOP 5 pizza_name, SUM(total_price)
AS Total_Revenue FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue DESC
```



	pizza_name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Spicy Italian Pizza	34831.25

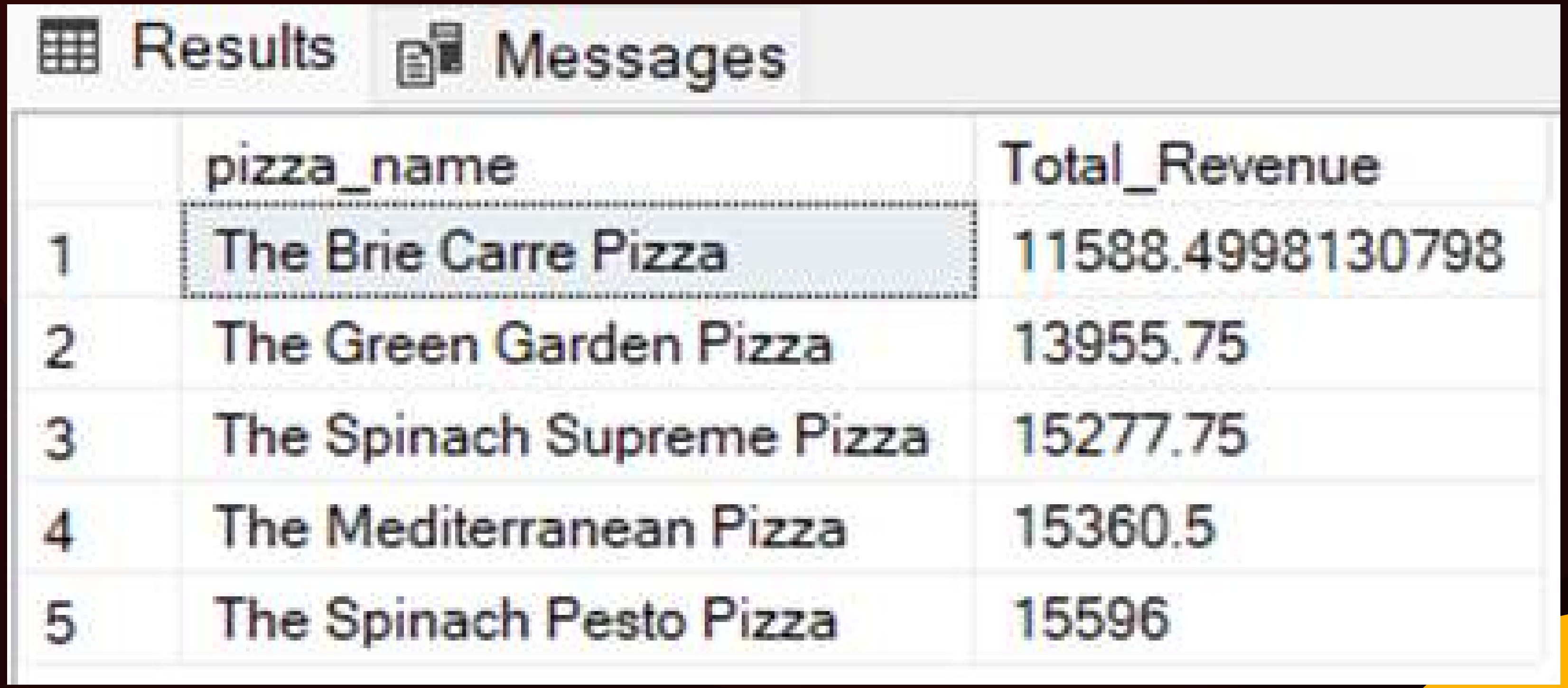
PROBLEM STATEMENT

06 BOTTOM 5 BEST SELLER BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS:

Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will enable us to identify underperforming or less popular pizza options.

OUTPUT :

```
SELECT TOP 5 pizza_name,
SUM(total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC
```



	pizza_name	Total_Revenue
1	The Brie Carre Pizza	11588.4998130798
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.5
5	The Spinach Pesto Pizza	15596

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 :

01

```
SELECT DATENAME(DW, order_date) as order_day, COUNT(DISTINCT order_id) AS Total_orders from pizza_sales
WHERE MONTH(order_date) = 1
GROUP BY DATENAME(DW, order_date)
```

*Here MONTH(order_date) = 1 indicates that the output is for month of January.
MONTH(order_date) = 4 indicates output for month of April.

02

```
SELECT DATENAME(DW, order_date) as order_day, COUNT(DISTINCT order_id) AS Total_orders from pizza_sales
WHERE DATEPART(quarter, order_date)=1
GROUP BY DATENAME(DW, order_date)
```

*Here DATEPART(quarter, order_date)=1 indicates that the output is for Quarter 1.
MONTH(order_date) = 3 indicates output for month of Quarter 3.

SUMMARY

The Pizza Sales Analysis project uses SQL to analyze and interpret pizza sales data to provide insights into sales performance and customer behavior. Key findings from the query and analysis of the dataset include identifying the most popular pizzas, peak sales times, and revenue trends. The project will show areas of strength, such as best-selling pizza types, and areas that need improvement, such as underperforming menu items or low-sales periods.

The analysis further reveals customer preference patterns that will be used to drive data-based decisions on improving inventory, refining marketing activities, and improving operational efficiency. This project demonstrates how SQL can be a powerful tool for deriving meaningful insights from raw data, contributing to improved business strategies and increased profitability in the competitive food industry.

THANK YOU

FOR ATTENTION

Mayuri Pachpor.