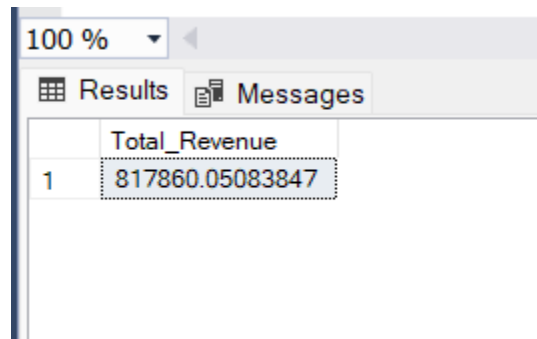


PIZZA SALES SQL QUERIES

- KPI
- Total Revenue

```
SELECT SUM(total_price) AS Total_Revenue from  
pizza_sales
```

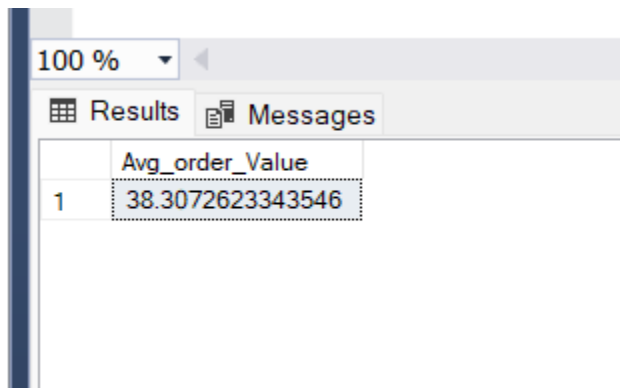


A screenshot of a SQL query results window. At the top, there is a zoom level dropdown set to '100 %' and two tabs: 'Results' (active) and 'Messages'. The results table has one column labeled 'Total_Revenue' and one row with the value '817860.05083847'.

	Total_Revenue
1	817860.05083847

- Average Order Value

```
SELECT *FROM pizza_sales  
SELECT (SUM(total_price) / COUNT(DISTINCT order_id)) AS  
Avg_order_Value FROM pizza_sales
```



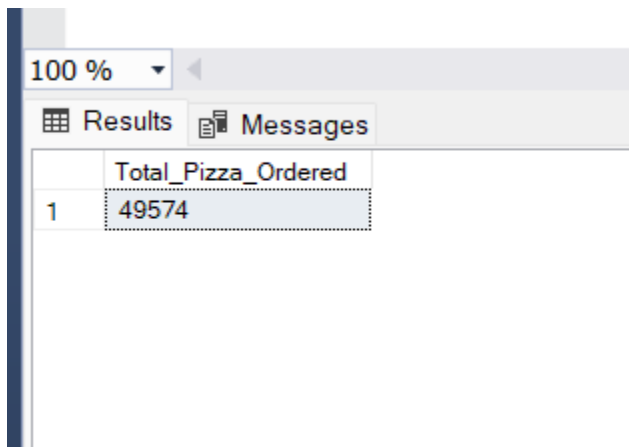
A screenshot of a SQL query results window. At the top, there is a zoom level dropdown set to '100 %' and two tabs: 'Results' (active) and 'Messages'. The results table has one column labeled 'Avg_order_Value' and one row with the value '38.3072623343546'.

	Avg_order_Value
1	38.3072623343546

- Total Pizza Sold

```
SELECT *FROM pizza_sales
```

```
SELECT SUM(quantity) as Total_Pizza_Ordered from  
pizza_sales
```



The screenshot shows a SQL Server query results window. At the top, there is a zoom level dropdown set to '100 %' and a scroll bar. Below these are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single-row table with two columns: 'Total_Pizza_Ordered' and its value '49574'.

	Total_Pizza_Ordered
1	49574

- Total Orders

```
SELECT *FROM pizza_sale
```

```
SELECT Count(DISTINCT order_id)from pizza_sale
```

100 %

Results Messages

	(No column name)
1	21350

- Average Pizza per order

```
SELECT *FROM pizza_sale
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_sale
```

100 %

Results

Messages

	Avg_Pizzas_per_order
1	2.32

- Weekly Order Day Trend

```
SELECT DATENAME(DW, order_date) AS order_day,
COUNT(DISTINCT order_id) as Total_Order from pizza_sale
GROUP BY DATENAME(DW, order_date)
ORDER BY Total_Order DESC
```

100 %

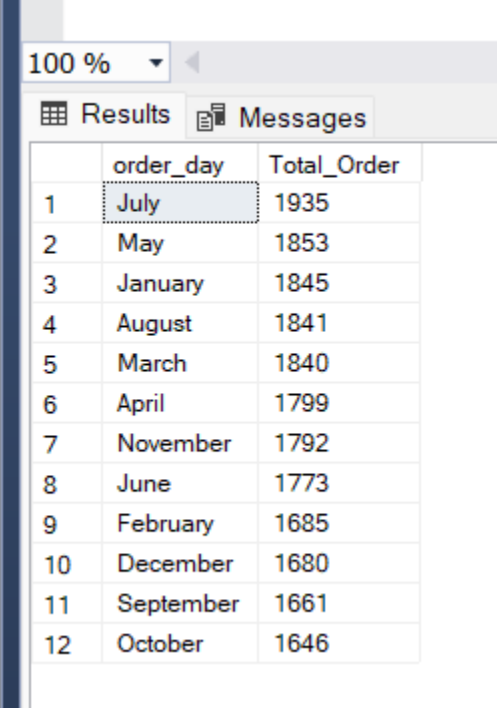
Results		Messages
	order_day	Total_Order
1	Friday	3538
2	Thursday	3239
3	Saturday	3158
4	Wednesday	3024
5	Tuesday	2973
6	Monday	2794
7	Sunday	2624

- Monthly Order Trend

```

SELECT DATENAME(MONTH, order_date) AS order_day,
COUNT(DISTINCT order_id) as Total_Order from pizza_sale
GROUP BY DATENAME(MONTH, order_date)
ORDER BY Total_Order DESC

```



The screenshot shows a SQL query results window with a zoom level of 100%. It contains two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

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

- Percentage Sales By Pizza Category

```

SELECT pizza_category, 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 Percentage_Total_Sales
FROM pizza_sale
GROUP BY pizza_category
ORDER BY total_revenue DESC

```

100 %

Results Messages

	pizza_category	total_revenue	Percentage_Total_Sales
1	Classic	220053.10	26.91
2	Supreme	208197.00	25.46
3	Chicken	195919.50	23.96
4	Veggie	193690.45	23.68

- Percentage Sales By Pizza Size

```

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 Percentage_Total_Sales
FROM pizza_sale
GROUP BY pizza_size
ORDER BY total_revenue DESC

```

100 %

Results Messages

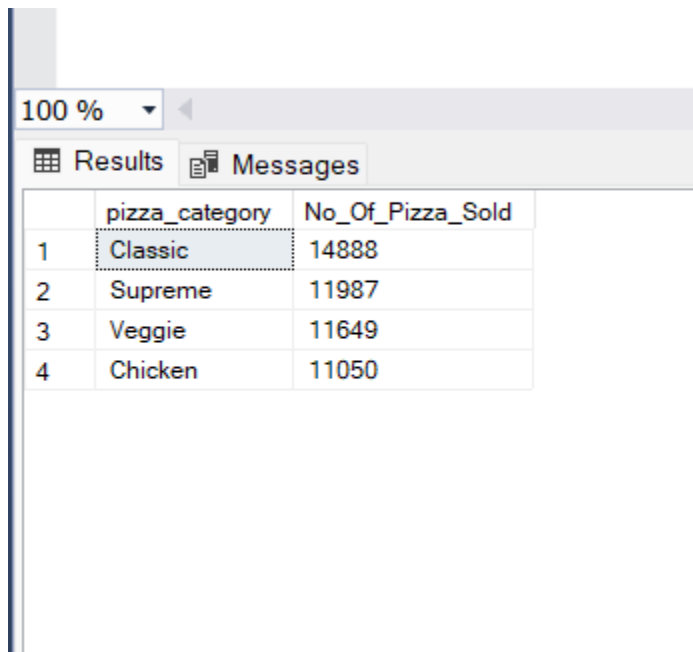
	pizza_size	total_revenue	Percentage_Total_Sales
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

- No of Pizza Sold By Category

```
SELECT pizza_category, SUM(quantity) as No_Of_Pizza_Sold FROM  
pizza_sale
```

```
GROUP BY pizza_category
```

```
ORDER BY No_Of_Pizza_Sold DESC
```



	pizza_category	No_Of_Pizza_Sold
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

- Top 5 Pizza by Revenue

```
SELECT TOP 5 pizza_name, SUM(total_price) AS Total_Revenue  
from pizza_sale
```

```
Group By pizza_name
```

```
ORDER By Total_Revenue DESC
```

Results Messages		
	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

- Bottom 5 Pizza by Revenue

SELECT TOP 5 pizza_name, SUM(total_price) AS Total_Revenue
from pizza_sale

Group By pizza_name

ORDER By Total_Revenue ASC

Results Messages		
	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

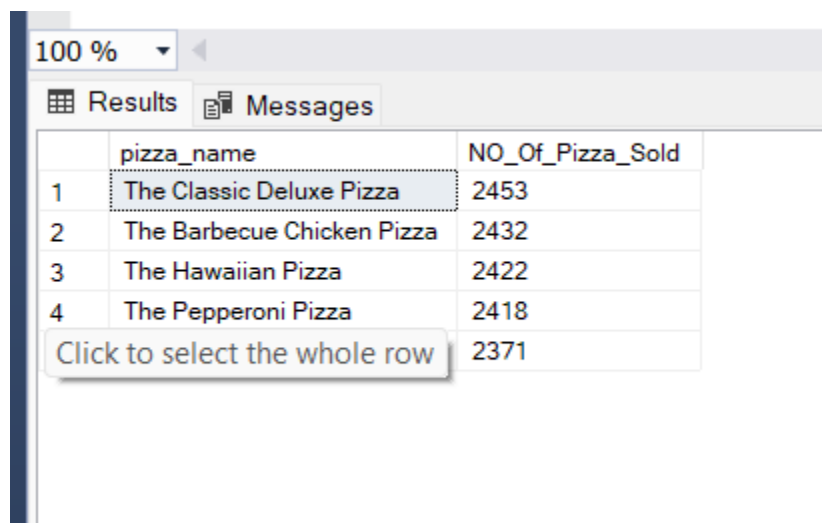
- **Top 5 Pizzas by Quantity**

SELECT TOP 5 pizza_name, SUM(quantity) AS

NO_Of_Pizza_Sold from pizza_sale

Group By pizza_name

ORDER By NO_Of_Pizza_Sold DESC



	pizza_name	NO_Of_Pizza_Sold
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
		2371

- **Bottom 5 Pizzas by Quantity**

SELECT TOP 5 pizza_name, SUM(quantity) AS

NO_Of_Pizza_Sold from pizza_sale

Group By pizza_name

ORDER By NO_Of_Pizza_Sold ASC

100 %

Results Messages

	pizza_name	NO_Of_Pizza_Sold
1	The Brie Carre Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961

• Top 5 Pizzas by Total Orders

SELECT Top 5 pizza_name, Count(DISTINCT order_id) as No_of_order
from pizza_sale

Group By pizza_name

ORDER By No_of_order DESC

100 %

Results Messages

	pizza_name	No_of_order
1	The Classic Deluxe Pizza	2329
2	The Hawaiian Pizza	2280
3	The Pepperoni Pizza	2278
4	The Barbecue Chicken Pizza	2273
5	The Thai Chicken Pizza	2225

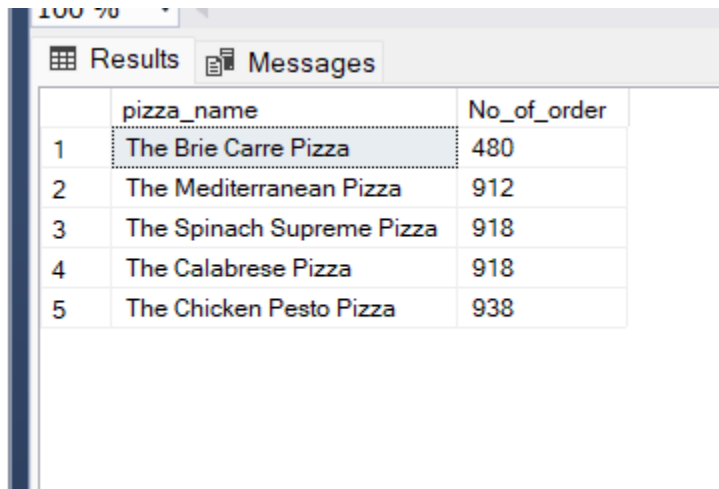
- **Bottom 5 Pizzas by Total Orders**

SELECT Top 5 pizza_name, Count(DISTINCT order_id) as

No_of_order from pizza_sale

Group By pizza_name

ORDER By No_of_order ASC



The screenshot shows a database query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'pizza_name' and 'No_of_order'. The table contains five rows of data, ordered by the number of orders from lowest to highest. The first row is 'The Brie Carre Pizza' with 480 orders. The second row is 'The Mediterranean Pizza' with 912 orders. The third row is 'The Spinach Supreme Pizza' with 918 orders. The fourth row is 'The Calabrese Pizza' with 918 orders. The fifth row is 'The Chicken Pesto Pizza' with 938 orders.

	pizza_name	No_of_order
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Spinach Supreme Pizza	918
4	The Calabrese Pizza	918
5	The Chicken Pesto Pizza	938

