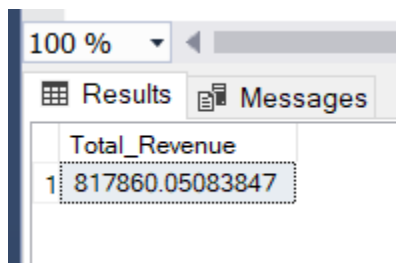


# Pizza sales : Query

## A: KPIs

### 1: Total Revenue:

```
SELECT SUM(total_price) as Total_Revenue from pizza_sales;
```

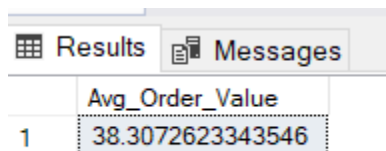


A screenshot of a SQL query results window. At the top, there is a zoom level dropdown set to '100 %' and a scroll bar. Below these are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single row with the column header 'Total\_Revenue' and the value '817860.05083847'.

	Total_Revenue
1	817860.05083847

### 2: Average Order Value:

```
SELECT SUM(total_price)/COUNT(distinct(order_id)) as 'Avg_Order_Value' from pizza_sales;
```

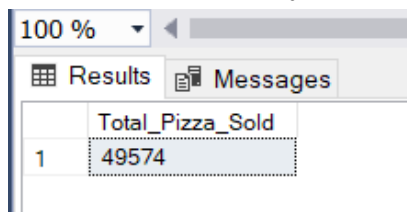


A screenshot of a SQL query results window. At the top, there are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single row with the column header 'Avg\_Order\_Value' and the value '38.3072623343546'.

	Avg_Order_Value
1	38.3072623343546

### 3: Total pizzas sold

```
SELECT SUM(quantity) as 'Total_Pizza_Sold' from pizza_sales;
```



A screenshot of a SQL 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 with the column header 'Total\_Pizza\_Sold' and the value '49574'.

	Total_Pizza_Sold
1	49574

#### 4: Total Orders

```
SELECT COUNT(DISTINCT(order_id)) AS 'Total Orders' FROM pizza_sales;
```

100 %

Results Messages

	Total Orders
1	21350

#### 5. Average order value:

```
SELECT SUM(total_price)/ COUNT(DISTINCT ORDER_id) as 'avg order value' from  
pizza_sales
```

100 %

Results Messages

	avg order value
1	38.3072623343546

#### 6. Total pizzas sold:

```
SELECT SUM(quantity) as 'Total pizzas sold' from pizza_sales
```

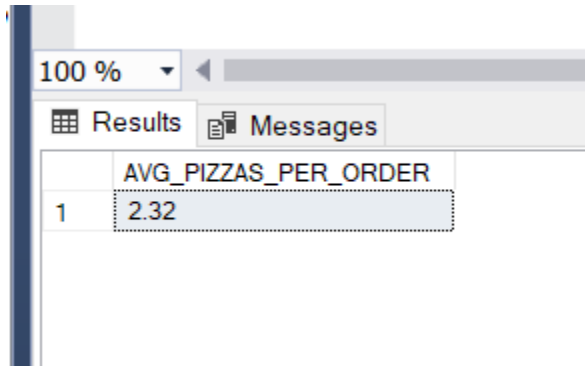
100 %

Results Messages

	Total pizzas sold
1	49574

## 7. 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_PIZZAS_PER_ORDER'  
from pizza_sales;
```

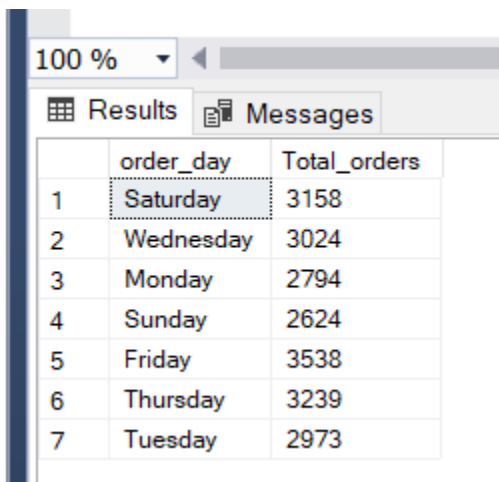


The screenshot shows a SQL Server query results window. The 'Results' tab is active, displaying a single row with the column name 'AVG\_PIZZAS\_PER\_ORDER' and the value '2.32'. The window has a zoom level of 100% and a scrollbar.

	AVG_PIZZAS_PER_ORDER
1	2.32

## 8. Daily trends 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);
```

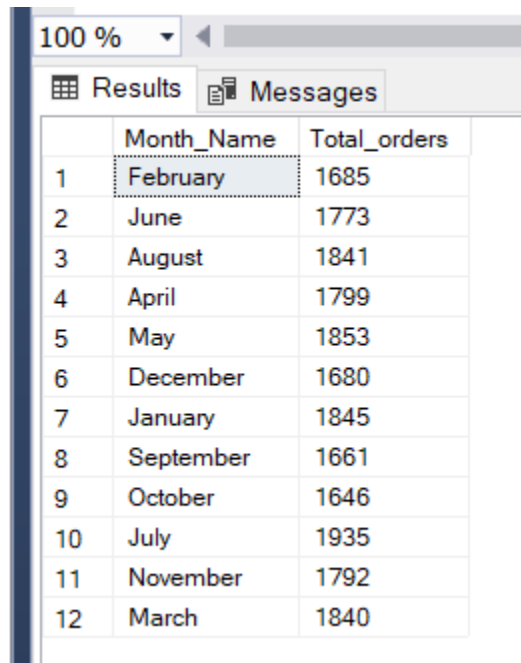


The screenshot shows a SQL Server query results window. The 'Results' tab is active, displaying a table with two columns: 'order\_day' and 'Total\_orders'. The table has 7 rows, one for each day of the week. The window has a zoom level of 100% and a scrollbar.

	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

### 9. Monthly trends for total orders:

```
select DATENAME(MONTH,order_date) as Month_Name,  
count(DISTINCT order_id) as Total_orders  
from pizza_sales  
group by DATENAME(MONTH,order_date);
```

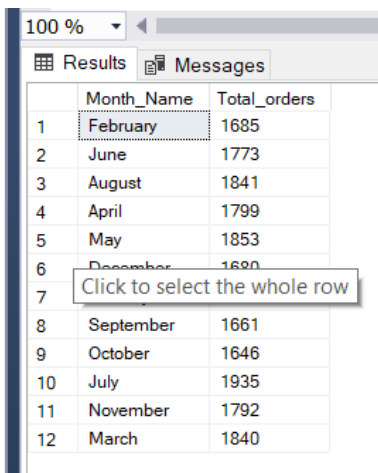


The screenshot shows a SQL Server query results window. The 'Results' tab is active, displaying a table with two columns: 'Month\_Name' and 'Total\_orders'. The table contains 12 rows, numbered 1 to 12. The first row is highlighted. The zoom level is set to 100%.

	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

For descending order:

```
select DATENAME(MONTH,order_date) as Month_Name,  
count(DISTINCT order_id) as Total_orders  
from pizza_sales  
group by DATENAME(MONTH,order_date)  
Order by total_orders DESC;
```

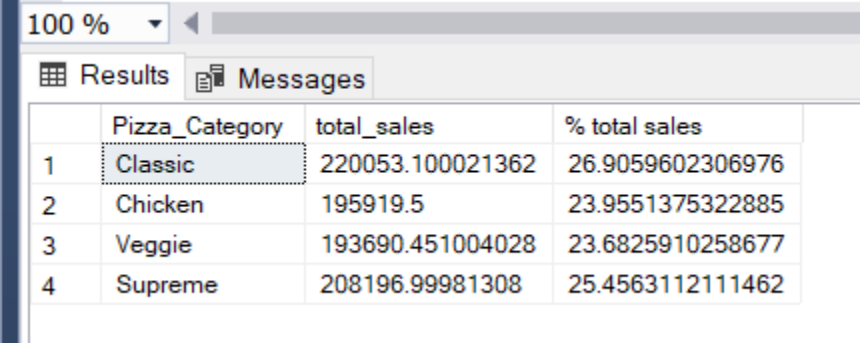


The screenshot shows a SQL Server query results window. The 'Results' tab is active, displaying a table with two columns: 'Month\_Name' and 'Total\_orders'. The table contains 12 rows, numbered 1 to 12. The first row is highlighted. A tooltip 'Click to select the whole row' is visible over the first row. The zoom level is set to 100%.

	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

10: % 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 '% total sales'  
from pizza_sales  
group by pizza_category;
```



The screenshot shows a SQL query results window with a zoom level of 100%. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with four columns: an index, 'Pizza\_Category', 'total\_sales', and '% total sales'. The table contains four rows of data for different pizza categories.

	Pizza_Category	total_sales	% total sales
1	Classic	220053.100021362	26.9059602306976
2	Chicken	195919.5	23.9551375322885
3	Veggie	193690.451004028	23.6825910258677
4	Supreme	208196.99981308	25.4563112111462

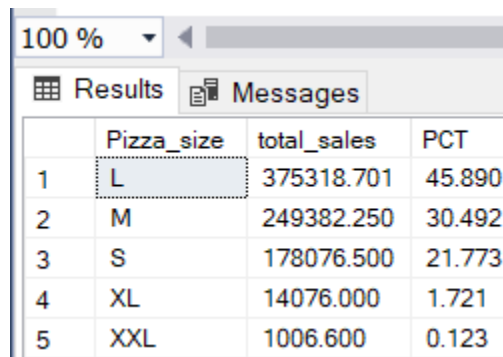
applying the where clause to get the monthly data:

```
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 '% total sales'  
from pizza_sales  
where MONTH(order_date)=1  
group by pizza_category;
```

–Note: month (order\_date)=1 means january

11. Percentage sales by pizza size: (WHERE clause for selecting quarter)

```
SELECT Pizza_size ,  
cast(sum(total_price) as decimal (10,3)) as total_sales ,  
cast(sum(total_price)*100/  
  (SELECT sum(total_price) from pizza_sales) as decimal(10,3)) as PCT  
from pizza_sales  
where DATEPART(quarter,order_date)=1  
group by pizza_size  
order by PCT desc;
```

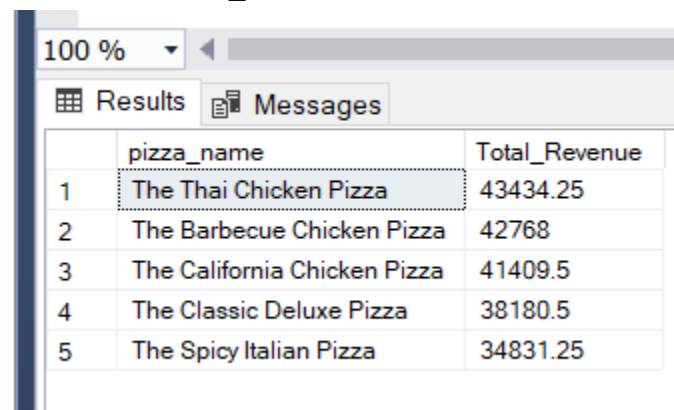


The screenshot shows a SQL Server query results window. At the top, there is a zoom level of 100% and a scroll bar. Below the zoom bar are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with four columns: an index column, 'Pizza\_size', 'total\_sales', and 'PCT'. The table contains five rows of data, sorted by 'PCT' in descending order. The first row (index 1) shows 'L' size with a total sales of 375318.701 and a percentage of 45.890. The second row (index 2) shows 'M' size with a total sales of 249382.250 and a percentage of 30.492. The third row (index 3) shows 'S' size with a total sales of 178076.500 and a percentage of 21.773. The fourth row (index 4) shows 'XL' size with a total sales of 14076.000 and a percentage of 1.721. The fifth row (index 5) shows 'XXL' size with a total sales of 1006.600 and a percentage of 0.123.

	Pizza_size	total_sales	PCT
1	L	375318.701	45.890
2	M	249382.250	30.492
3	S	178076.500	21.773
4	XL	14076.000	1.721
5	XXL	1006.600	0.123

12: top 5 pizza by total revenue:

```
SELECT TOP 5 pizza_name, sum(total_price) AS Total_Revenue  
from pizza_sales  
group by pizza_name  
ORDER BY Total_Revenue desc;
```

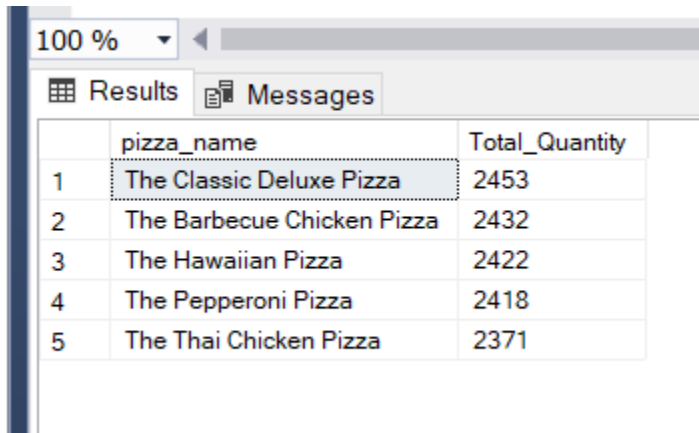


The screenshot shows a SQL Server query results window. At the top, there is a zoom level of 100% and a scroll bar. Below the zoom bar are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: an index column and two data columns, 'pizza\_name' and 'Total\_Revenue'. The table contains five rows of data, sorted by 'Total\_Revenue' in descending order. The first row (index 1) shows 'The Thai Chicken Pizza' with a total revenue of 43434.25. The second row (index 2) shows 'The Barbecue Chicken Pizza' with a total revenue of 42768. The third row (index 3) shows 'The California Chicken Pizza' with a total revenue of 41409.5. The fourth row (index 4) shows 'The Classic Deluxe Pizza' with a total revenue of 38180.5. The fifth row (index 5) shows 'The Spicy Italian Pizza' with a total revenue of 34831.25.

	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

13: top 5 pizza 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;
```

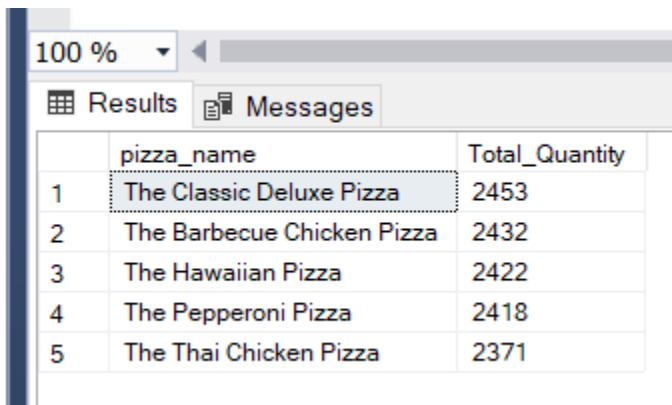


The screenshot shows a SQL Server query results window. At the top, there is a zoom level of 100% and a scroll bar. Below the zoom level are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with two columns: 'pizza\_name' and 'Total\_Quantity'. The table contains five rows, numbered 1 to 5, representing the top 5 pizzas by total quantity. The first row is 'The Classic Deluxe Pizza' with a total quantity of 2453. The second row is 'The Barbecue Chicken Pizza' with a total quantity of 2432. The third row is 'The Hawaiian Pizza' with a total quantity of 2422. The fourth row is 'The Pepperoni Pizza' with a total quantity of 2418. The fifth row is 'The Thai Chicken Pizza' with a total quantity of 2371.

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

14: top 5 pizza by total\_orders :

```
SELECT TOP 5 pizza_name, sum(quantity) AS Total_Quantity
from pizza_sales
group by pizza_name
ORDER BY Total_Quantity desc;
```



The screenshot shows a SQL Server query results window. At the top, there is a zoom level of 100% and a scroll bar. Below the zoom level are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with two columns: 'pizza\_name' and 'Total\_Quantity'. The table contains five rows, numbered 1 to 5, representing the top 5 pizzas by total quantity. The first row is 'The Classic Deluxe Pizza' with a total quantity of 2453. The second row is 'The Barbecue Chicken Pizza' with a total quantity of 2432. The third row is 'The Hawaiian Pizza' with a total quantity of 2422. The fourth row is 'The Pepperoni Pizza' with a total quantity of 2418. The fifth row is 'The Thai Chicken Pizza' with a total quantity of 2371.

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