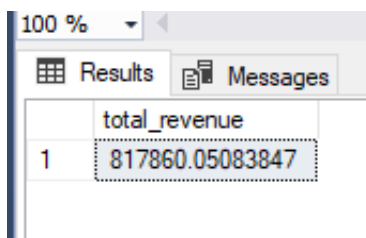


# Sql query

## 1. KPI

- **Total revenue**

`SELECT sum(total_price) as total_revenue FROM pizza_sales;`

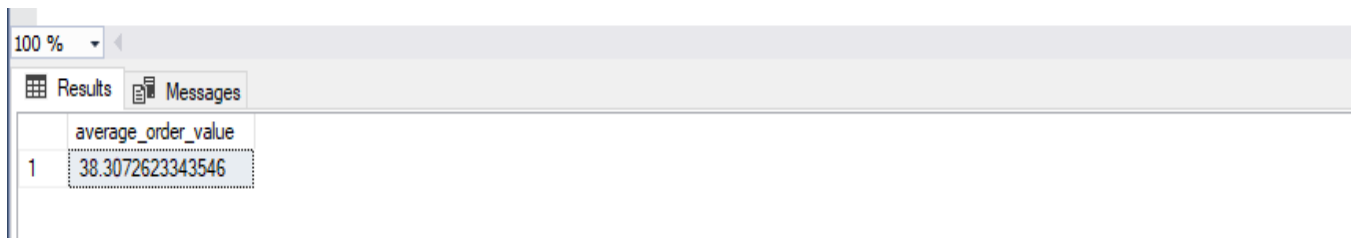


A screenshot of a SQL Server query results window. The window has a tab labeled 'Results' and a 'Messages' tab. The zoom level is set to 100%. The results are displayed in a table with one column named 'total\_revenue' and one row containing the value 817860.05083847.

	total_revenue
1	817860.05083847

- **Average order value**

`select (sum(total_price)/count(distinct order_id)) as average_order_value  
from pizza_sales;`

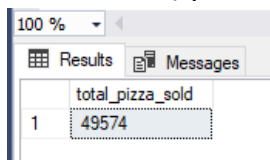


A screenshot of a SQL Server query results window. The window has a tab labeled 'Results' and a 'Messages' tab. The zoom level is set to 100%. The results are displayed in a table with one column named 'average\_order\_value' and one row containing the value 38.3072623343546.

	average_order_value
1	38.3072623343546

- **Total pizza sold**

`select sum(quantity) as total_pizza_sold from pizza_sales;`

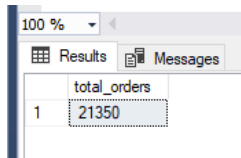


A screenshot of a SQL Server query results window. The window has a tab labeled 'Results' and a 'Messages' tab. The zoom level is set to 100%. The results are displayed in a table with one column named 'total\_pizza\_sold' and one row containing the value 49574.

	total_pizza_sold
1	49574

- **Total orders**

```
select count(distinct (order_id)) as total_orders from pizza_sales;
```

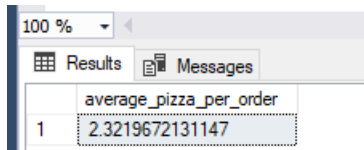


A screenshot of a SQL Server query results window. The window has a title bar with '100 %' and a scroll bar. Below the title bar are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab shows a single row with two columns: 'total\_orders' and the value '21350'.

	total_orders
1	21350

- **Average pizza per order**

```
select (cast (sum (quantity) as decimal (10,2))/  
cast (count (distinct order_id) as decimal (10,2))) as average_pizza_per_order  
from pizza_sales;
```



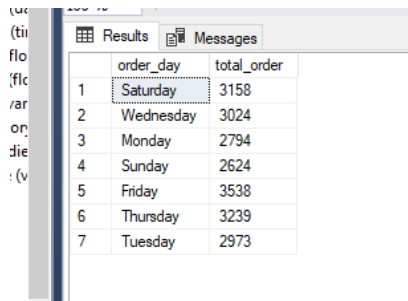
A screenshot of a SQL Server query results window. The window has a title bar with '100 %' and a scroll bar. Below the title bar are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab shows a single row with two columns: 'average\_pizza\_per\_order' and the value '2.3219672131147'.

	average_pizza_per_order
1	2.3219672131147

## 2. Charts

- B. 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);
```

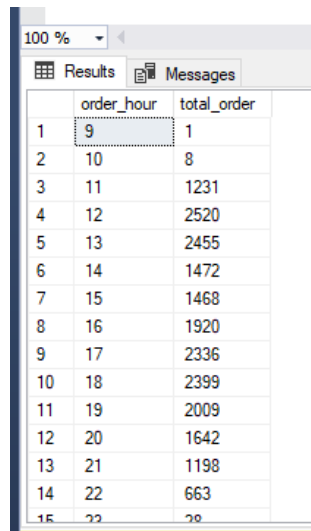


A screenshot of a SQL Server query window showing the results of a query. The window has tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'order\_day' and 'total\_order'. The table contains seven rows of data, representing the days of the week. The 'order\_day' column lists the days from Saturday to Tuesday, and the 'total\_order' column shows the corresponding total number of orders for each day.

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

- C. Hourly Trend for Orders**

```
select datepart (hour,order_time) as order_hour , count ( distinct order_id) as  
total_order from pizza_sales  
group by datepart (hour,order_time)  
order by datepart (hour,order_time);
```

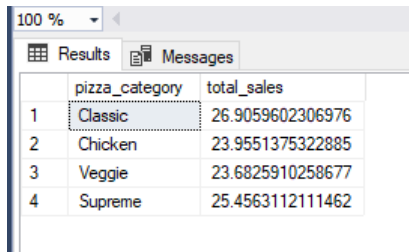


A screenshot of a SQL Server query window showing the results of a query. The window has tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'order\_hour' and 'total\_order'. The table contains 15 rows of data, representing the hours of the day from 9 to 23. The 'order\_hour' column lists the hours, and the 'total\_order' column shows the corresponding total number of orders for each hour.

	order_hour	total_order
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

- **D. % of Sales by Pizza Category**

```
select pizza_category, sum(total_price)*100/(select sum(total_price) from
pizza_sales) as total_sales
from pizza_sales
group by pizza_category;
```

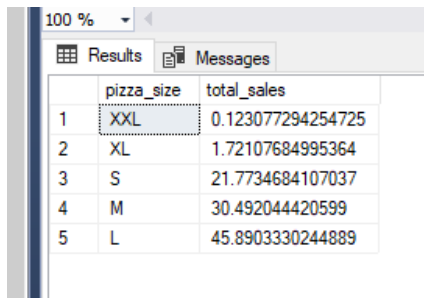


A screenshot of a SQL query results window. The window has a dropdown menu at the top set to '100 %'. Below the menu are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with two columns: 'pizza\_category' and 'total\_sales'. The table contains four rows of data, numbered 1 to 4. The first row is 'Classic' with a value of 26.9059602306976. The second row is 'Chicken' with a value of 23.9551375322885. The third row is 'Veggie' with a value of 23.6825910258677. The fourth row is 'Supreme' with a value of 25.4563112111462.

	pizza_category	total_sales
1	Classic	26.9059602306976
2	Chicken	23.9551375322885
3	Veggie	23.6825910258677
4	Supreme	25.4563112111462

- **E. % of Sales by Pizza Size**

```
select pizza_size, sum(total_price)*100/(select sum(total_price) from pizza_sales)
as total_sales
from pizza_sales
group by pizza_size
order by total_sales;
```

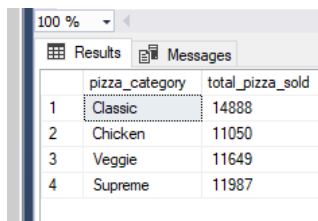


A screenshot of a SQL query results window. The window has a dropdown menu at the top set to '100 %'. Below the menu are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with two columns: 'pizza\_size' and 'total\_sales'. The table contains five rows of data, numbered 1 to 5. The first row is 'XXL' with a value of 0.123077294254725. The second row is 'XL' with a value of 1.72107684995364. The third row is 'S' with a value of 21.7734684107037. The fourth row is 'M' with a value of 30.492044420599. The fifth row is 'L' with a value of 45.8903330244889.

	pizza_size	total_sales
1	XXL	0.123077294254725
2	XL	1.72107684995364
3	S	21.7734684107037
4	M	30.492044420599
5	L	45.8903330244889

- **F. Total Pizzas Sold by Pizza Category**

```
select pizza_category, sum (quantity) as total_pizza_sold from pizza_sales
group by pizza_category;
```



A screenshot of a SQL query results window. The window has a dropdown menu at the top set to '100 %'. Below the menu are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with two columns: 'pizza\_category' and 'total\_pizza\_sold'. The table contains four rows of data, numbered 1 to 4. The first row is 'Classic' with a value of 14888. The second row is 'Chicken' with a value of 11050. The third row is 'Veggie' with a value of 11649. The fourth row is 'Supreme' with a value of 11987.

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

- **Top 5 Best Sellers by Total Pizzas Sold**

```
select pizza_name, sum (quantity) as total_pizza_sales
from pizza_sales
group by pizza_name
order by total_pizza_sales desc;
```

Results		Messages
	pizza_name	total_pizza_sales
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
6	The California Chicken Pizza	2370

- **Bottom 5 Best Sellers by Total Pizzas Sold**

```
select pizza_name, sum (quantity) as total_pizza_sales
from pizza_sales
group by pizza_name
order by total_pizza_sales;
```

	pizza_name	total_pizza_sales
1	The Brie Carré Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961
6	The Spinach Pesto Pizza	970
7	The Chicken Pesto Pizza	973
8	The Italian Vegetables Pizza	981
9	The Chicken Alfredo Pizza	987