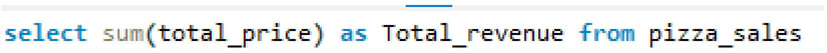
SQL queries

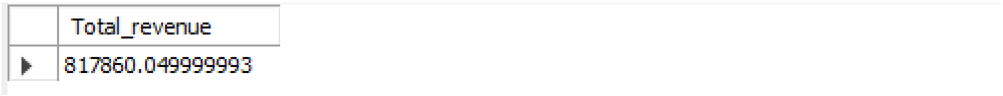
Requirements

KPI

1. **Total Revenue:** The sum of total price of all pizza orders

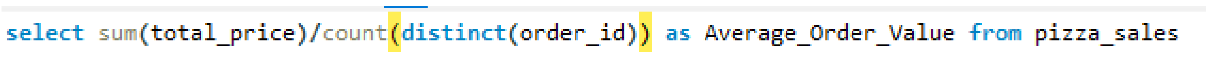


```
select sum(total_price) as Total_revenue from pizza_sales
```

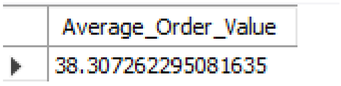


	Total_revenue
▶	817860.049999993

2. **Average Order Value:** The average amount spent per order, calculated by dividing total revenue by total number of orders

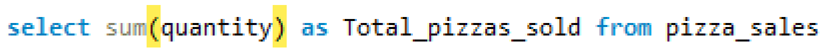


```
select sum(total_price)/count(distinct(order_id)) as Average_Order_Value from pizza_sales
```

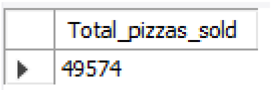


	Average_Order_Value
▶	38.307262295081635

3. **Total pizzas sold:** The sum of quantities of all pizzas sold

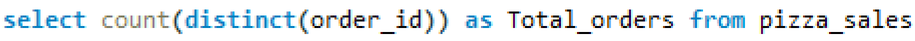


```
select sum(quantity) as Total_pizzas_sold from pizza_sales
```

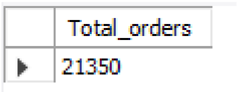


	Total_pizzas_sold
▶	49574

4. **Total Orders:** The total number of orders placed

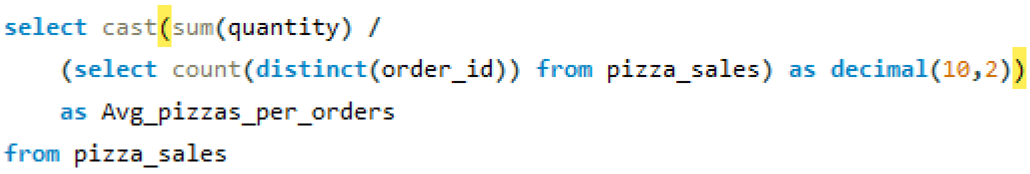


```
select count(distinct(order_id)) as Total_orders from pizza_sales
```

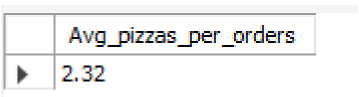


	Total_orders
▶	21350

5. **Average Pizzas Per Order:** Calculated by dividing total number of pizzas sold by the total number of orders



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

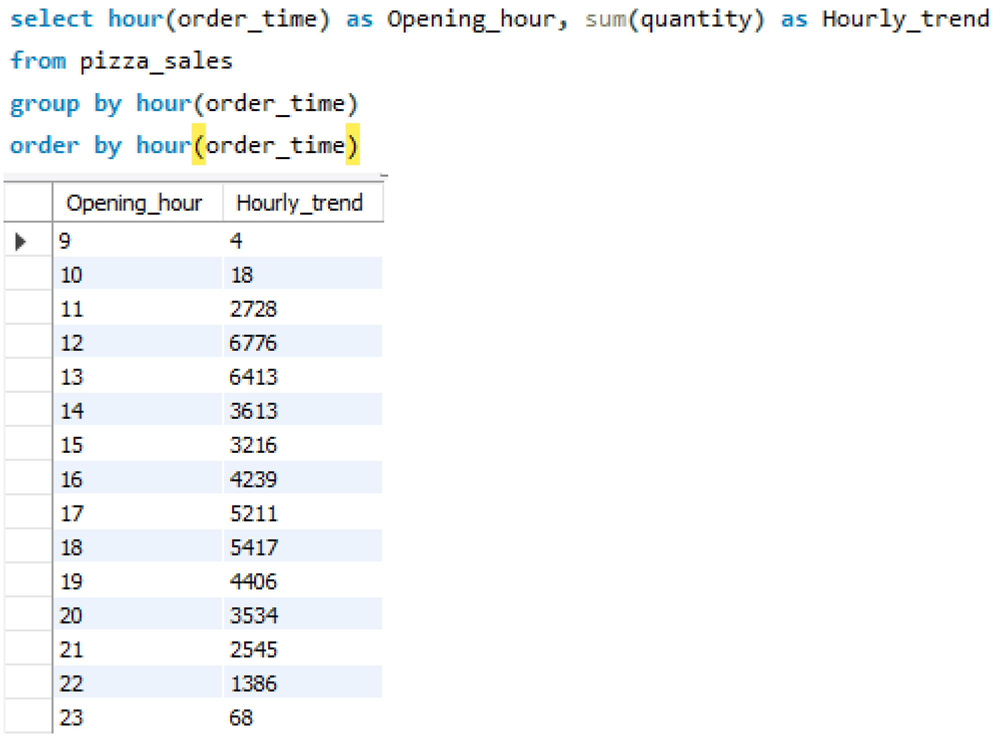


	Avg_pizzas_per_orders
▶	2.32

CHARTS

1. Hourly trend for total orders

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

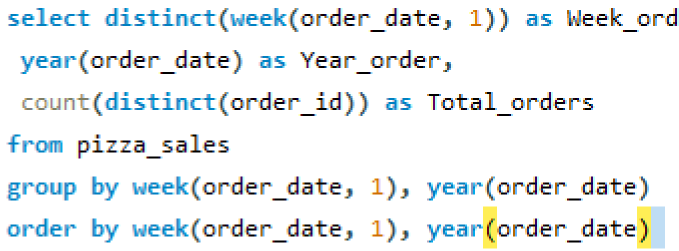


```sql
select hour(order_time) as Opening_hour, sum(quantity) as Hourly_trend
from pizza_sales
group by hour(order_time)
order by hour(order_time)
```

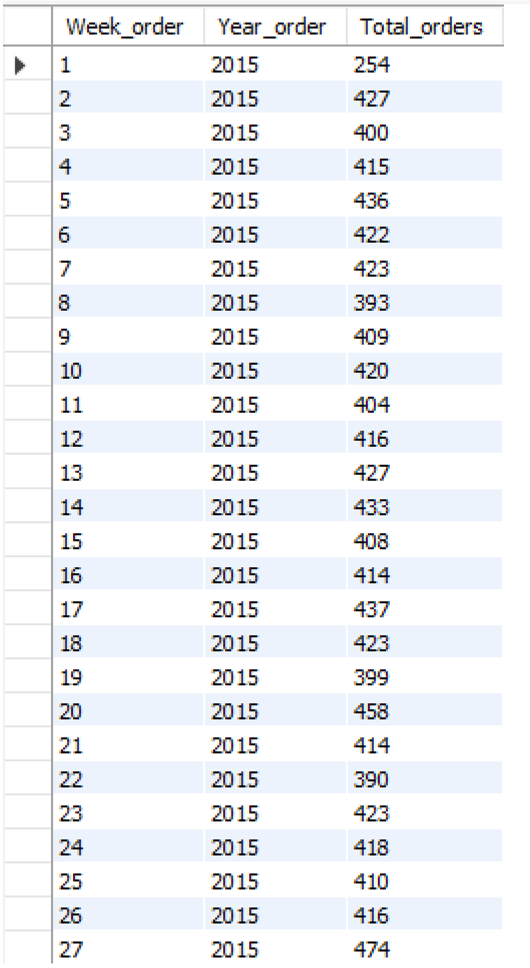
Opening_hour	Hourly_trend
9	4
10	18
11	2728
12	6776
13	6413
14	3613
15	3216
16	4239
17	5211
18	5417
19	4406
20	3534
21	2545
22	1386
23	68

2. Weekly trend for total orders

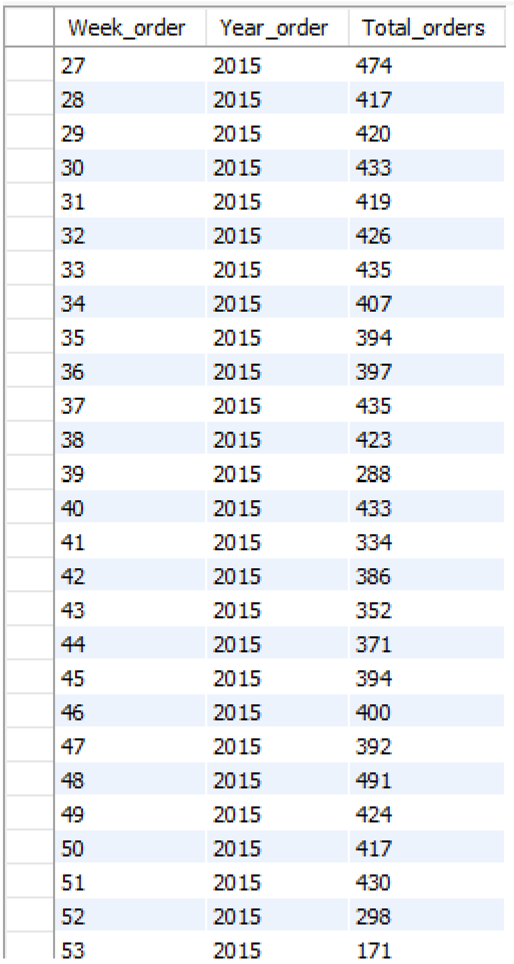
- Create a line chart that illustrates the weekly trend of total orders throughout the year. This chart will allow us to identify peak weeks or periods of high order activity.



```sql
select distinct(week(order_date, 1)) as Week_order,
 year(order_date) as Year_order,
 count(distinct(order_id)) as Total_orders
from pizza_sales
group by week(order_date, 1), year(order_date)
order by week(order_date, 1), year(order_date)
```



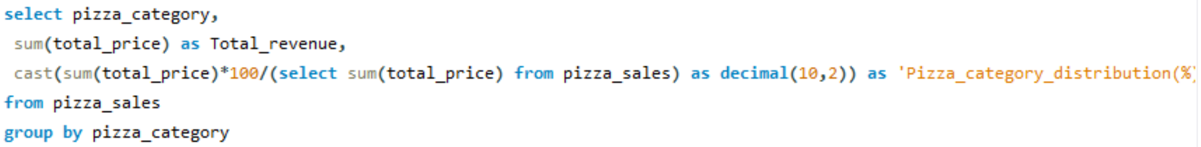
	Week_order	Year_order	Total_orders
▶	1	2015	254
	2	2015	427
	3	2015	400
	4	2015	415
	5	2015	436
	6	2015	422
	7	2015	423
	8	2015	393
	9	2015	409
	10	2015	420
	11	2015	404
	12	2015	416
	13	2015	427
	14	2015	433
	15	2015	408
	16	2015	414
	17	2015	437
	18	2015	423
	19	2015	399
	20	2015	458
	21	2015	414
	22	2015	390
	23	2015	423
	24	2015	418
	25	2015	410
	26	2015	416
	27	2015	474



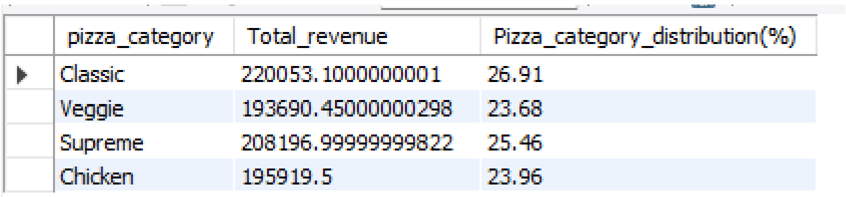
	Week_order	Year_order	Total_orders
	27	2015	474
	28	2015	417
	29	2015	420
	30	2015	433
	31	2015	419
	32	2015	426
	33	2015	435
	34	2015	407
	35	2015	394
	36	2015	397
	37	2015	435
	38	2015	423
	39	2015	288
	40	2015	433
	41	2015	334
	42	2015	386
	43	2015	352
	44	2015	371
	45	2015	394
	46	2015	400
	47	2015	392
	48	2015	491
	49	2015	424
	50	2015	417
	51	2015	430
	52	2015	298
	53	2015	171

3. Percentage 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.



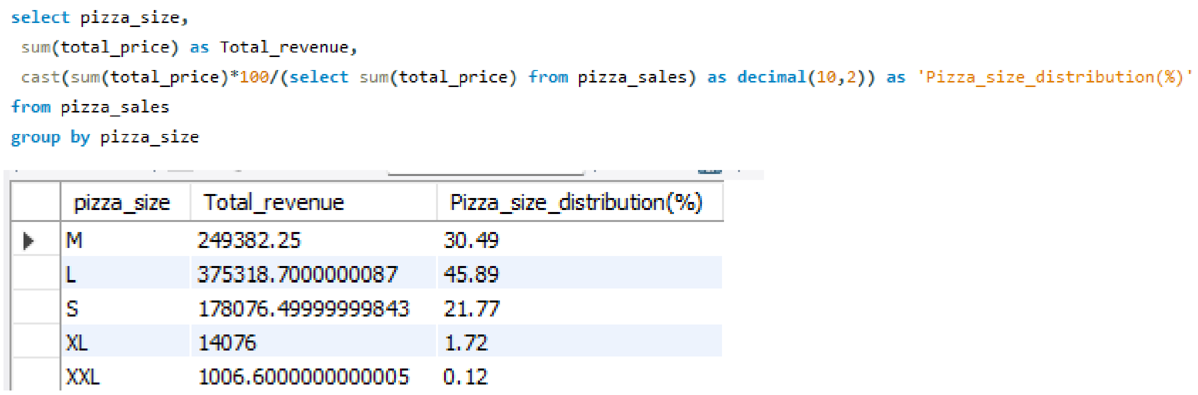
```
select pizza_category,
 sum(total_price) as Total_revenue,
 cast(sum(total_price)*100/(select sum(total_price) from pizza_sales) as decimal(10,2)) as 'Pizza_category_distribution(%
from pizza_sales
group by pizza_category
```



	pizza_category	Total_revenue	Pizza_category_distribution(%)
▶	Classic	220053.1000000001	26.91
	Veggie	193690.45000000298	23.68
	Supreme	208196.99999999822	25.46
	Chicken	195919.5	23.96

4. Percentage of sale by Pizza size

- Generate a pie 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.

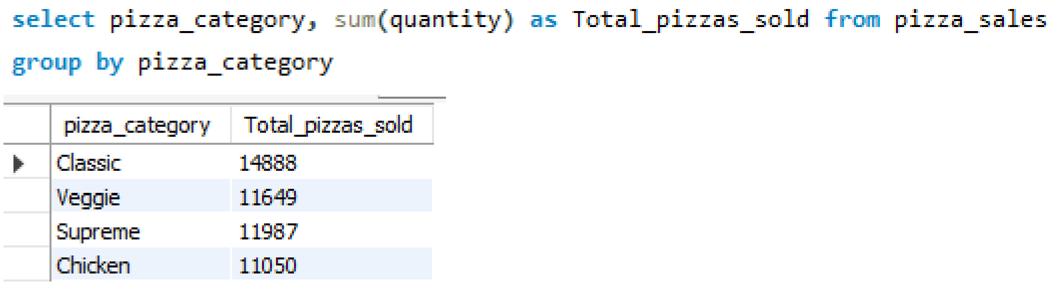


```
select pizza_size,
 sum(total_price) as Total_revenue,
 cast(sum(total_price)*100/(select sum(total_price) from pizza_sales) as decimal(10,2)) as 'Pizza_size_distribution(%)'
from pizza_sales
group by pizza_size
```

pizza_size	Total_revenue	Pizza_size_distribution(%)
M	249382.25	30.49
L	375318.7000000087	45.89
S	178076.49999999843	21.77
XL	14076	1.72
XXL	1006.6000000000005	0.12

5. Total pizzas sold by pizza category

- Create a funnel chart that presents the total number of pizzas sold for each pizza category. This chart will allow us to compare the sales performance of different pizza categories.



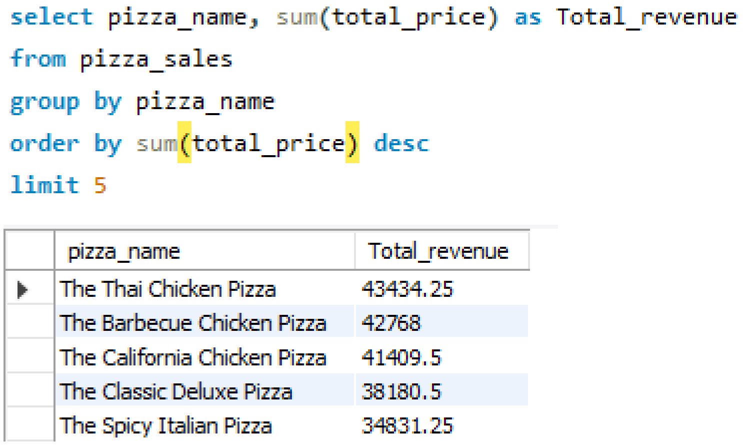
```
select pizza_category, sum(quantity) as Total_pizzas_sold from pizza_sales
group by pizza_category
```

pizza_category	Total_pizzas_sold
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

6. Top 5 best sellers 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.

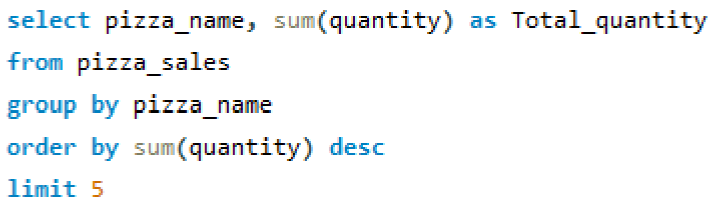
Top 5 best sellers by revenue:



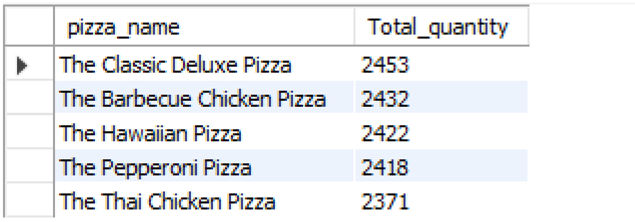
```
select pizza_name, sum(total_price) as Total_revenue
from pizza_sales
group by pizza_name
order by sum(total_price) desc
limit 5
```

pizza_name	Total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25

Top 5 best sellers by Total quantity:

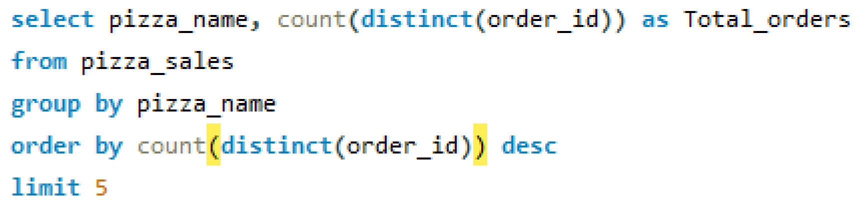


```
select pizza_name, sum(quantity) as Total_quantity
from pizza_sales
group by pizza_name
order by sum(quantity) desc
limit 5
```

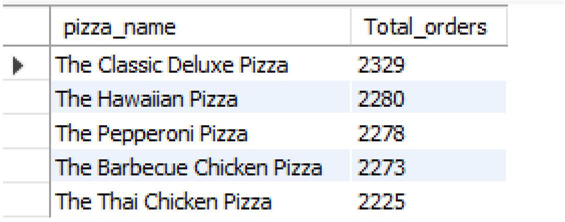


pizza_name	Total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Top 5 best sellers by Total orders:



```
select pizza_name, count(distinct(order_id)) as Total_orders
from pizza_sales
group by pizza_name
order by count(distinct(order_id)) desc
limit 5
```

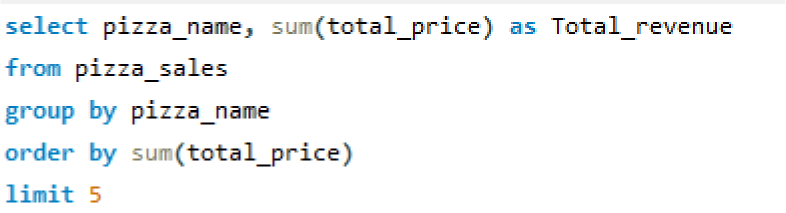


pizza_name	Total_orders
The Classic Deluxe Pizza	2329
The Hawaiian Pizza	2280
The Pepperoni Pizza	2278
The Barbecue Chicken Pizza	2273
The Thai Chicken Pizza	2225

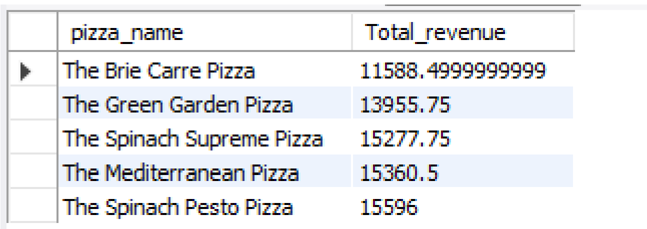
7. Bottom 5 best sellers by revenue, Total quantity and total orders

- 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.

Top 5 worst-selling by revenue:

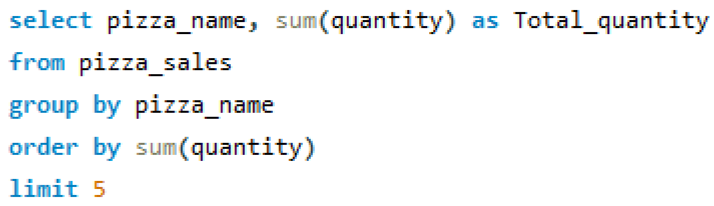


```
select pizza_name, sum(total_price) as Total_revenue
from pizza_sales
group by pizza_name
order by sum(total_price)
limit 5
```

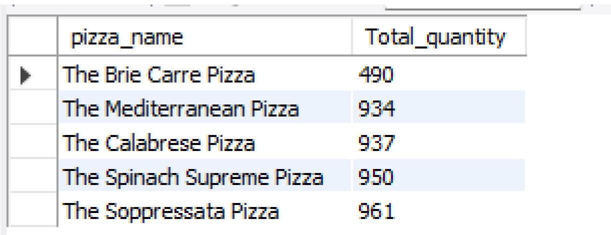


pizza_name	Total_revenue
The Brie Carre Pizza	11588.4999999999
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

Top 5 worst-selling by Total quantity:

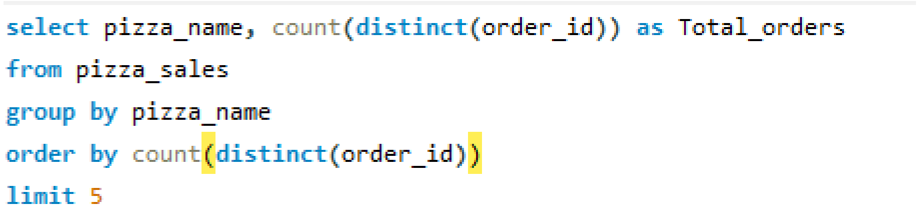


```
select pizza_name, sum(quantity) as Total_quantity
from pizza_sales
group by pizza_name
order by sum(quantity)
limit 5
```

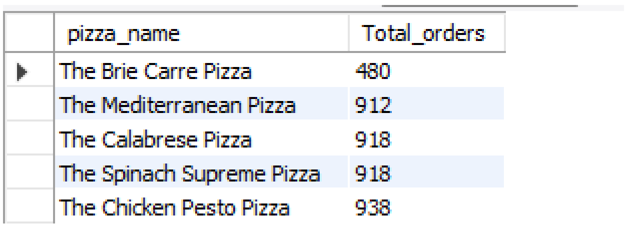


	pizza_name	Total_quantity
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961

Top 5 worst-selling by Total orders:



```
select pizza_name, count(distinct(order_id)) as Total_orders
from pizza_sales
group by pizza_name
order by count(distinct(order_id))
limit 5
```



	pizza_name	Total_orders
▶	The Brie Carre Pizza	480
	The Mediterranean Pizza	912
	The Calabrese Pizza	918
	The Spinach Supreme Pizza	918
	The Chicken Pesto Pizza	938