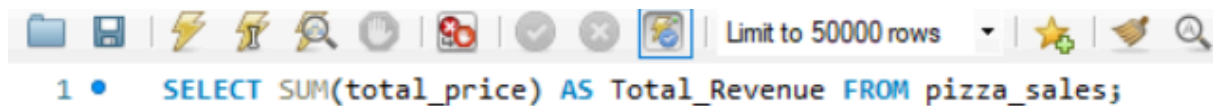


## PROBLEM STATEMENT

### 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:

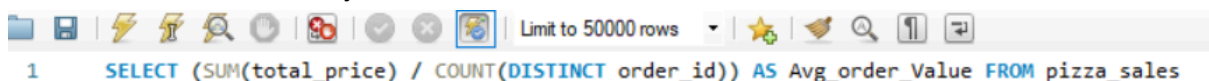
1. **Total Revenue:** The sum of the total price of all pizza orders



A screenshot of a SQL query editor interface. The toolbar at the top includes icons for file operations, search, and execution, along with a dropdown menu set to 'Limit to 50000 rows'. The SQL query entered is: `1 SELECT (SUM(total_price) / COUNT(DISTINCT order_id)) AS Avg_order_Value FROM pizza_sales`

Total_Revenue
817860.049999993

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



A screenshot of a SQL query editor interface. The toolbar at the top includes icons for file operations, search, and execution, along with a dropdown menu set to 'Limit to 50000 rows'. The SQL query entered is: `1 SELECT SUM(quantity) AS Total_Pizzas_Sold FROM pizza_sales`

Avg_order_Value
38.307262295081635

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

1 `SELECT SUM(quantity) AS Total_pizza_sold FROM pizza_sales`

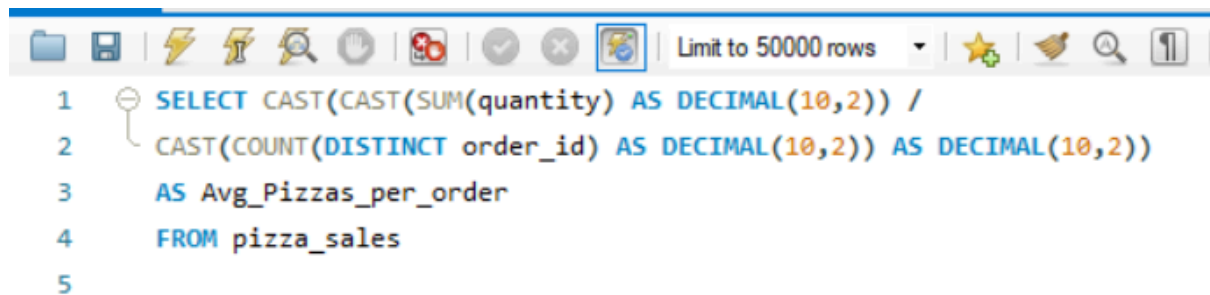
Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Total_pizza_sold			
▶	49574			

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

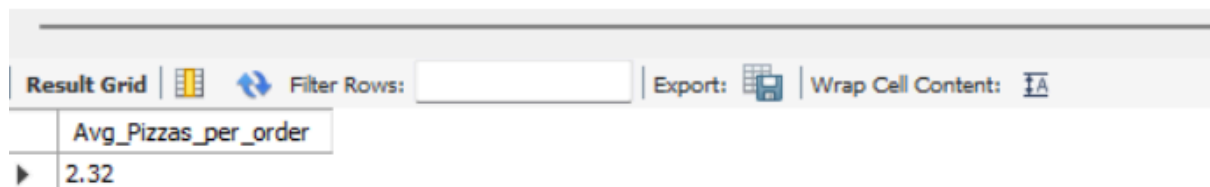
1 `SELECT COUNT(DISTINCT order_id) AS Total_Orders FROM pizza_sales`

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Total_Orders			
▶	21350			

5. **Average Pizza 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.



```
1 SELECT CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) /  
2 CAST(COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS DECIMAL(10,2))  
3 AS Avg_Pizzas_per_order  
4 FROM pizza_sales  
5
```



Avg_Pizzas_per_order
2.32

## Charts Requirement

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

### 1. 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.

1	•	SELECT DAYNAME(order_date) AS order_day, COUNT(DISTINCT order_id) AS total_orders
2		FROM pizza_sales
3		GROUP BY DAYNAME(order_date)

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
order_day	total_orders		
Friday	3538		
Monday	2794		
Saturday	3158		
Sunday	2624		
Thursday	3239		
Tuesday	2973		
Wednesday	3024		

## 2. 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.

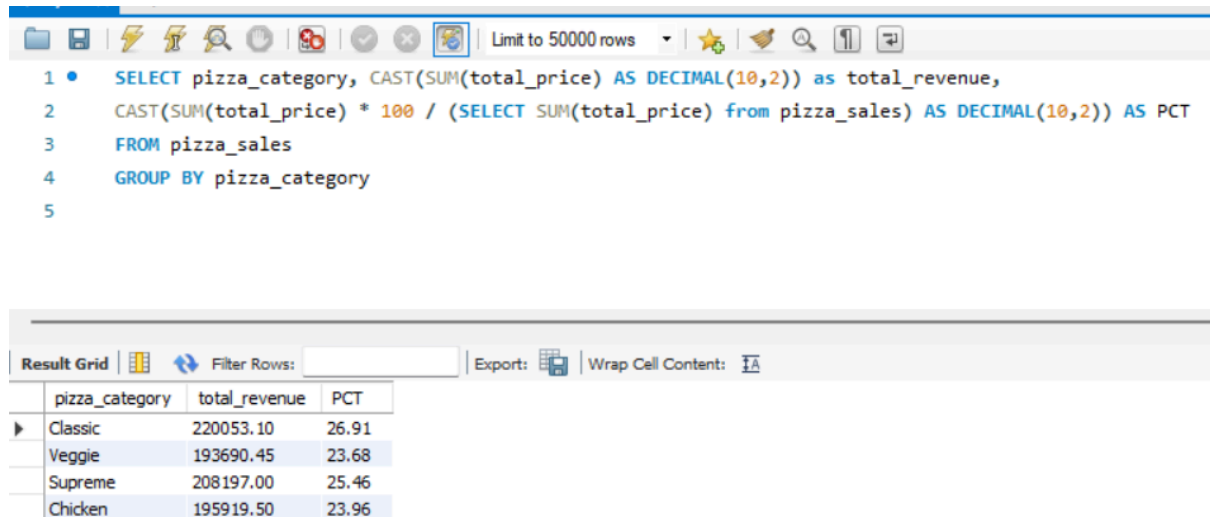
1	•	select MONTHNAME(order_date) as Month_Name, COUNT(DISTINCT order_id) as Total_Orders
2		from pizza_sales
3		GROUP BY MONTHNAME(order_date)

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Month_Name	Total_Orders		
April	1799		
August	1841		
December	1680		
February	1685		
January	1845		
July	1935		
June	1773		
March	1840		
May	1853		
November	1792		
October	1646		
September	1661		

### 3. Percentage of Sales by Pizza Category:

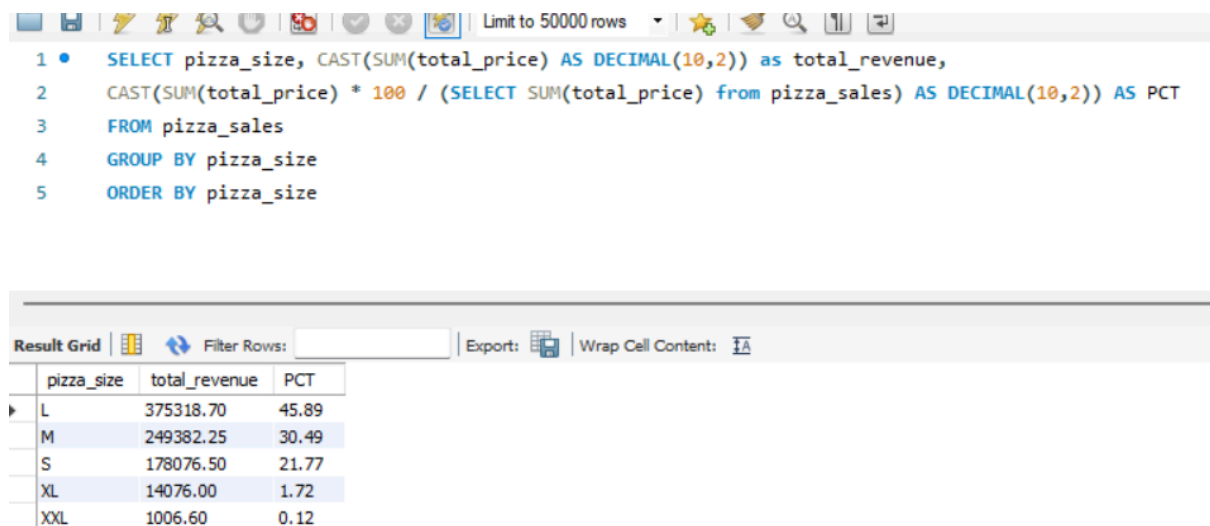
Create a pie chart that shows the distribution of sales across different pizza categories. This chart will provide insights into the popularity of various pizza categories and their contribution to overall sales.



```
1 • SELECT pizza_category, CAST(SUM(total_price) AS DECIMAL(10,2)) as total_revenue,
2     CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS DECIMAL(10,2)) AS PCT
3 FROM pizza_sales
4 GROUP BY pizza_category
5
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
pizza_category	total_revenue	PCT	
Classic	220053.10	26.91	
Veggie	193690.45	23.68	
Supreme	208197.00	25.46	
Chicken	195919.50	23.96	

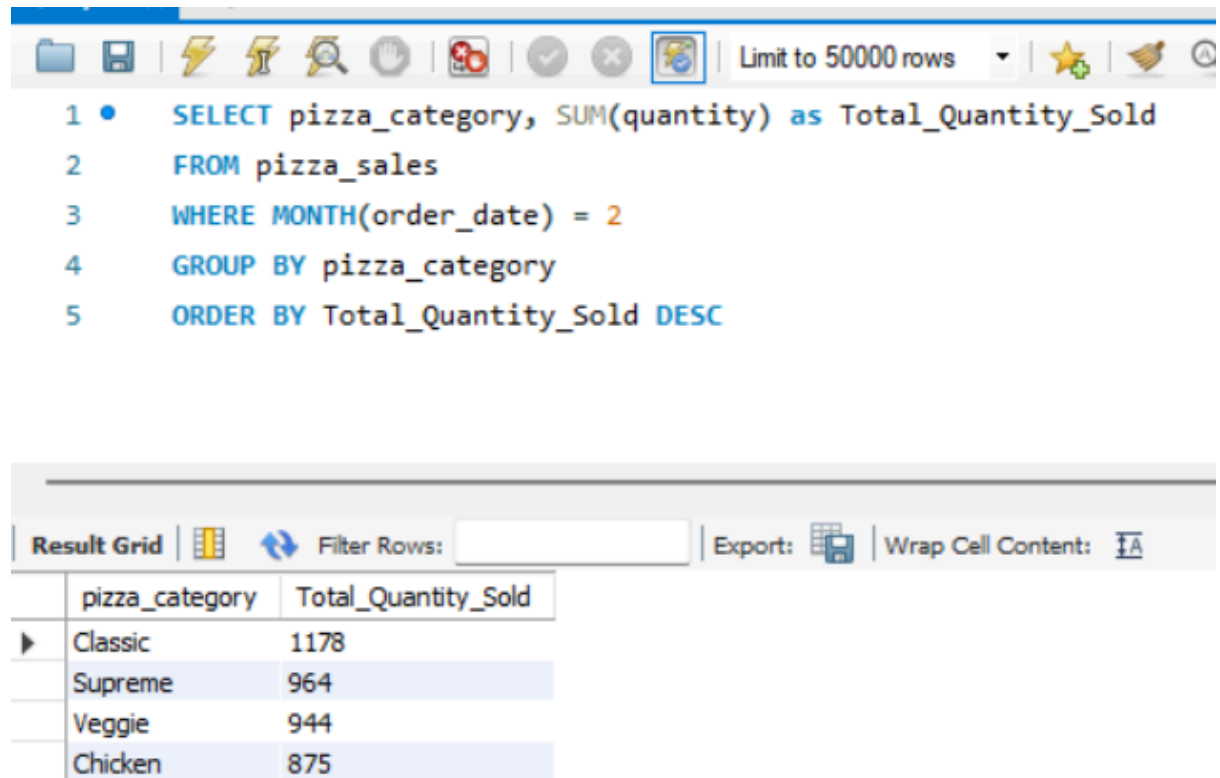
### Percentage of sales by pizza size:



```
1 • SELECT pizza_size, CAST(SUM(total_price) AS DECIMAL(10,2)) as total_revenue,
2     CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS DECIMAL(10,2)) AS PCT
3 FROM pizza_sales
4 GROUP BY pizza_size
5 ORDER BY pizza_size
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
pizza_size	total_revenue	PCT	
L	375318.70	45.89	
M	249382.25	30.49	
S	178076.50	21.77	
XL	14076.00	1.72	
XXL	1006.60	0.12	

### Total Pizzas Sold by Pizza Category:



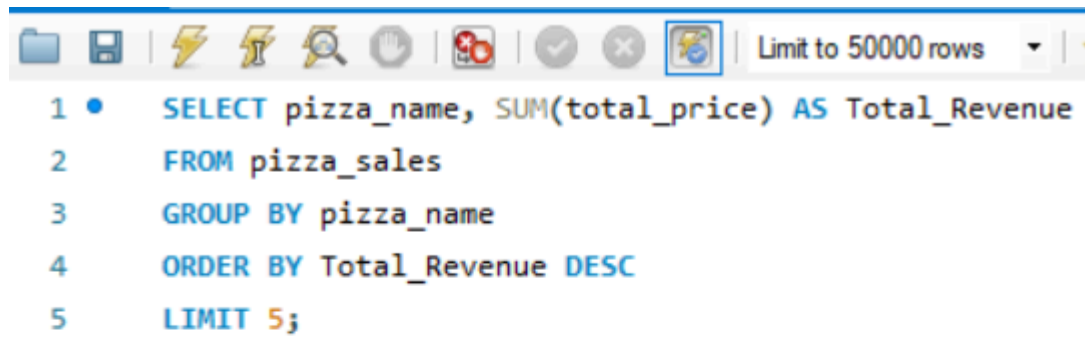
The image shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The query text is as follows:

```
1 • SELECT pizza_category, SUM(quantity) as Total_Quantity_Sold
2 FROM pizza_sales
3 WHERE MONTH(order_date) = 2
4 GROUP BY pizza_category
5 ORDER BY Total_Quantity_Sold DESC
```

Below the query editor is the 'Result Grid' section. It features a toolbar with 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The results are displayed in a table with two columns: 'pizza\_category' and 'Total\_Quantity\_Sold'.

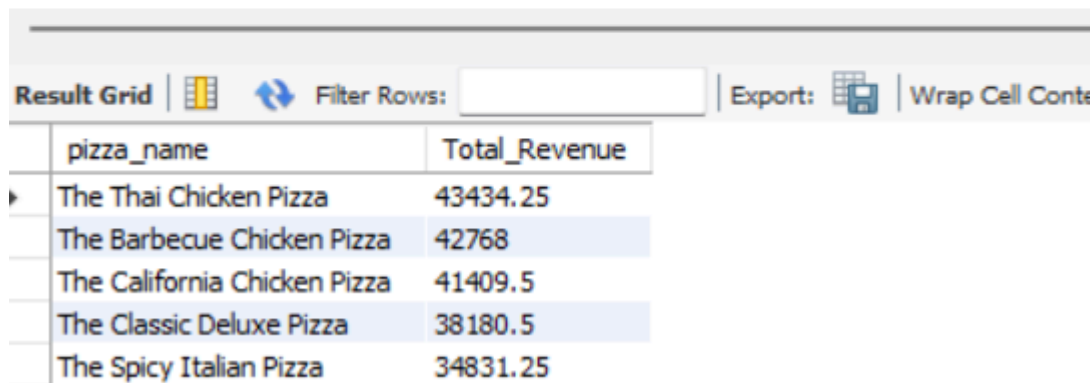
	pizza_category	Total_Quantity_Sold
▶	Classic	1178
	Supreme	964
	Veggie	944
	Chicken	875

### Top 5 Pizzas by Revenue:



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and search. The query text is as follows:


```
1 • SELECT pizza_name, SUM(total_price) AS Total_Revenue
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Revenue DESC
5 LIMIT 5;
```




The screenshot shows a 'Result Grid' interface with a toolbar for filtering, exporting, and wrapping text. The results are displayed in a table with two columns: 'pizza\_name' and 'Total\_Revenue'.

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

### Bottom 5 Pizzas by Revenue:



```
1 • SELECT pizza_name, SUM(total_price) AS Total_Revenue
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Revenue ASC
5 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 Pizzas by Quantity



```
1 • SELECT pizza_name, SUM(quantity) AS Total_Pizzas_Sold
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Pizzas_Sold DESC
5 LIMIT 5;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	pizza_name	Total_Pizzas_Sold			
▶	The Classic Deluxe Pizza	2453			
	The Barbecue Chicken Pizza	2432			
	The Hawaiian Pizza	2422			
	The Pepperoni Pizza	2418			
	The Thai Chicken Pizza	2371			

### Bottom 5 Pizzas by Quantity:



```
1 • SELECT pizza_name, SUM(quantity) AS Total_Pizzas_Sold
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Pizzas_Sold ASC
5 LIMIT 5;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	pizza_name	Total_Pizzas_Sold			
▶	The Brie Carre Pizza	490			
	The Mediterranean Pizza	934			
	The Calabrese Pizza	937			
	The Spinach Supreme Pizza	950			
	The Soppressata Pizza	961			

### Top 5 Pizzas by Total Orders:

```
1 • SELECT pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Orders DESC
5 LIMIT 5;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows
	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				

### Bottom 5 Pizzas by Total Orders:



```
1 • SELECT pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
2 FROM pizza_sales
3 GROUP BY pizza_name
4 ORDER BY Total_Orders ASC
5 LIMIT 5;
```

A screenshot of a database result grid. The toolbar at the top includes 'Result Grid', a grid icon, a refresh icon, a 'Filter Rows:' input field, an 'Export:' button, a 'Wrap Cell Content:' toggle, and a 'Fetch rows' button. The table below has two columns: 'pizza\_name' and 'Total\_Orders'. It lists the bottom 5 pizzas by total orders, with 'The Brie Carre Pizza' having the lowest total orders (480) and 'The Chicken Pesto Pizza' having the highest (938).

	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