OBJECTIVE

-To analyse customer traffic trends and identify peak hours in order to optimize staffing and improve operational efficiency.

-To identify the average number of pizzas in an order, as well as the top-selling pizzas, in order to inform inventory management and marketing strategies.

-To determine the total revenue generated by pizza sales throughout the year and identify any seasonal trends in order to inform forecasting and budgeting decisions.

-To assess the popularity of various pizzas on the menu and identify any underperforming items or opportunities for promotion in order to optimize the menu and increase sales.

About the dataset:

This dataset contains 12 columns.

- ✓ Pizza_id – It contains pizza id means unique id of pizza.
- ✓ Order_id - It contains the id by which pizza is ordered.
- ✓ Pizza_name_id – It contains name id of the particular pizza.
- ✓ Quantity – It is the no. of pizza sold.
- ✓ Order_date- It is the date on which pizza is ordered.
- ✓ Order_time- It is the time on which pizza is ordered.
- ✓ Unit_price – It is the price at which per unit sold.

- ✓ Total_price – It is the price of each order of pizza.
- ✓ Pizza_size- It is the size of pizza .Which is divided in 5 sizes .
- ✓ Pizza_category – The pizza is of different - different categories. i.e.

1. Classic
2. Supreme
3. Veggie
4. Chicken

- ✓ Pizza_ingredients- Ingredients used in pizza.
- ✓ Pizza_name- There are different – different names by which pizza is identified.

📝 Steps Performed:

There are following steps are performed:

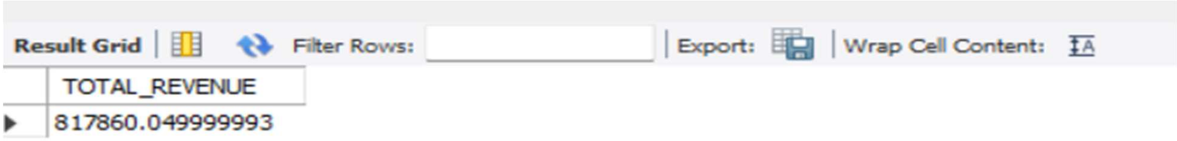
PROBLEM STATEMENT

KPI’S REQUIREMENT

We need to analyze 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**

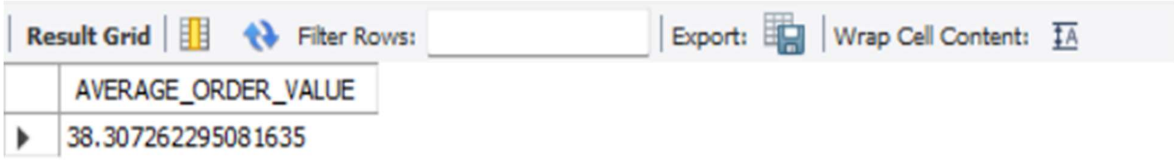
```
SELECT * FROM PIZZA_SALES;

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 the total revenue by the total number of orders**

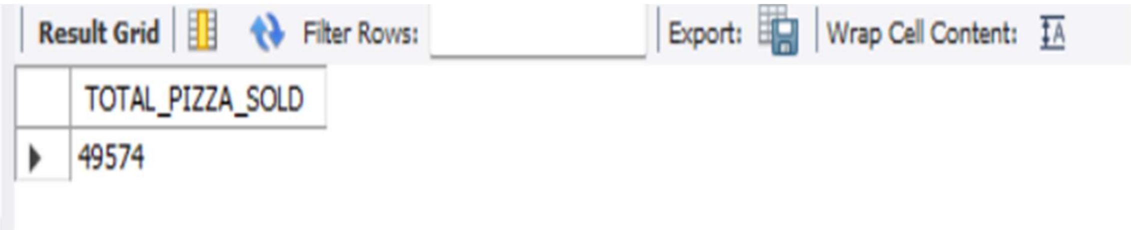
```
SELECT SUM (TOTAL_PRICE) / COUNT (DISTINCT ORDER_ID) AS
AVERAGE_ORDER_VALUE FROM PIZZA_SALES;
```



AVERAGE_ORDER_VALUE
38.30726229508 1635

3. **Total pizza sold = The sum of quantities of total pizza sold**

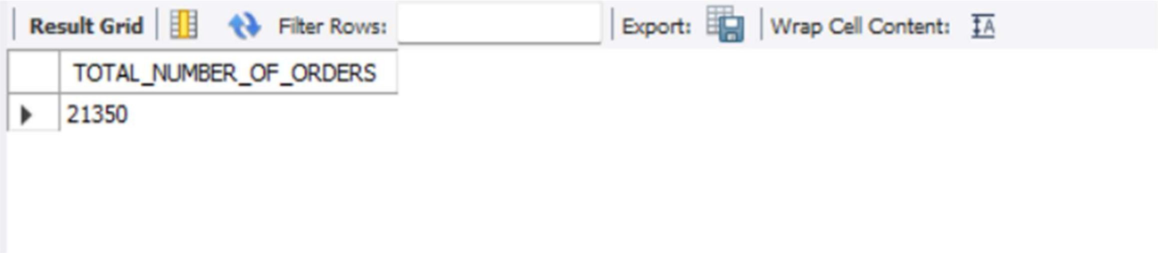
```
SELECT SUM (QUANTITY) AS TOTAL_PIZZA_SOLD FROM
PIZZA_SALES;
```



TOTAL_PIZZA_SOLD
49574

4. **Total order = total number of orders placed**

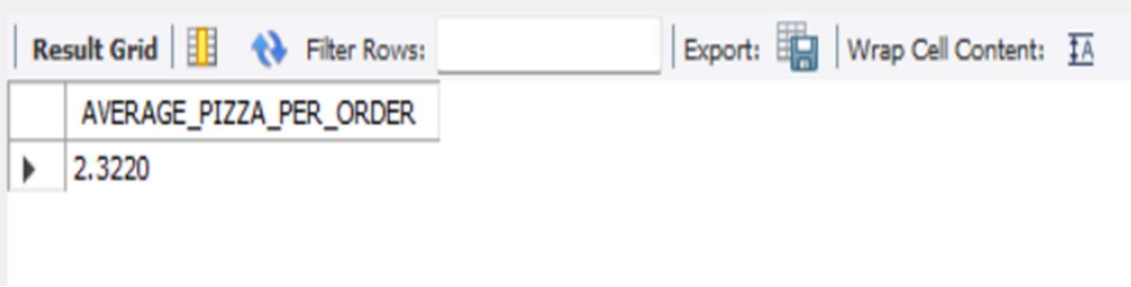
```
SELECT COUNT (DISTINCT ORDER_ID) AS
TOTAL_NUMBER_OF_ORDERS FROM PIZZA_SALES;
```



```
TOTAL_NUMBER_OF_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;**

```
SELECT SUM(QUANTITY)/COUNT (DISTINCT ORDER_ID) AS
AVERAGE_PIZZA_PER_ORDER FROM PIZZA_SALES;
```



```
AVERAGE_PIZZA_PER_ORDER
2.3220
```

CHARTS REQUIREMENTS

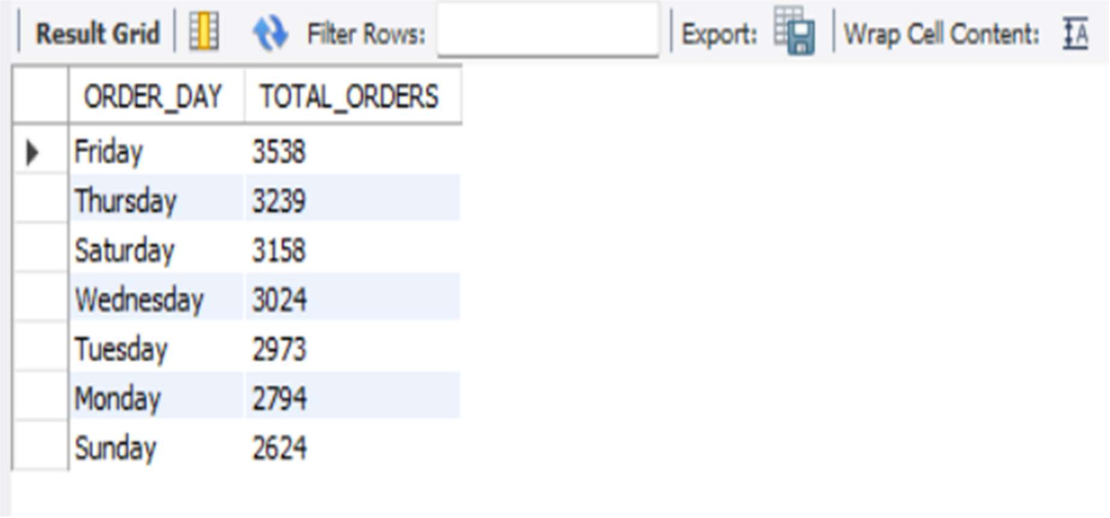
WE WOULD LIKE TO VISUALIZE VARIOUS ASPECTS OF OUR PIZZA SALES DATA TO GAIN INSIGHTS AND UNDERSTAND KEY TRENDS. WE HAVE IDENTIFIED

THE FOLLOWING REQUIREMENTS FOR CREATING CHARTS.

1. DAILY TRENDS 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.

```
SELECT DAYNAME(ORDER_DATE) AS ORDER_DAY,
COUNT (DISTINCT ORDER_ID) AS TOTAL_ORDERS,
FROM PIZZA_SALES
GROUP BY DAYNAME(ORDER_DATE)
ORDER BY TOTAL_ORDERS DESC;
```

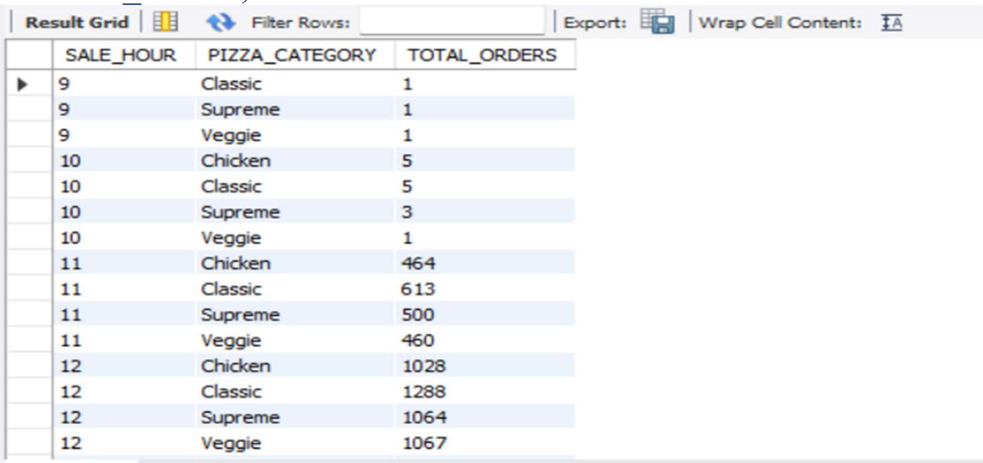


ORDER_DAY	TOTAL_ORDERS
Friday	3538
Thursday	3239
Saturday	3158
Wednesday	3024
Tuesday	2973
Monday	2794
Sunday	2624

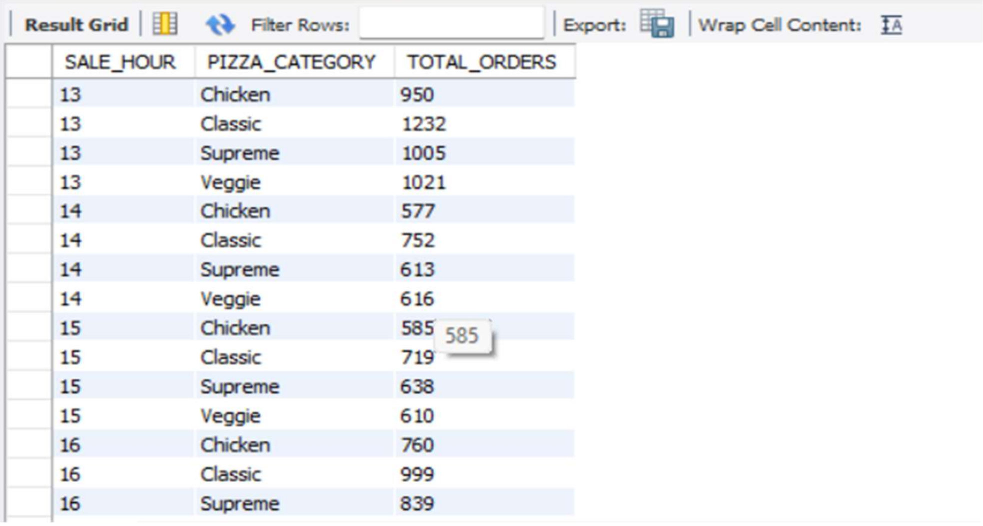
2. HOURLY TRENDS FOR TOTAL ORDERS

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.

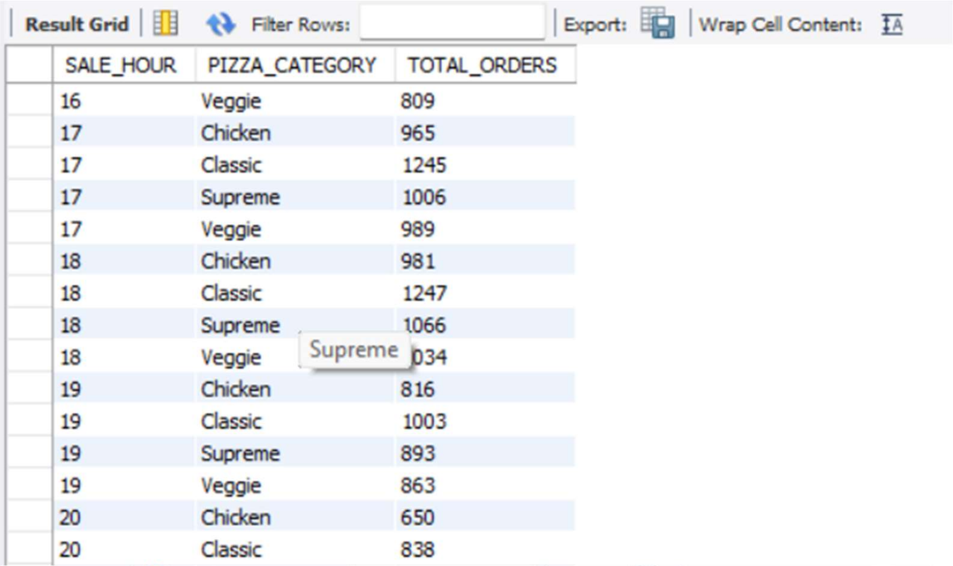
```
SELECT HOUR(ORDER_TIME) AS SALE HOUR(PIZZA_CATEGORY),
COUNT (DISTINCT ORDER_ID) AS TOTAL_ORDERS
FROM
   PIZZA_SALES
GROUP BY
   SALE_HOUR, PIZZA_CATEGORY
ORDER BY
   SALE_HOUR;
```



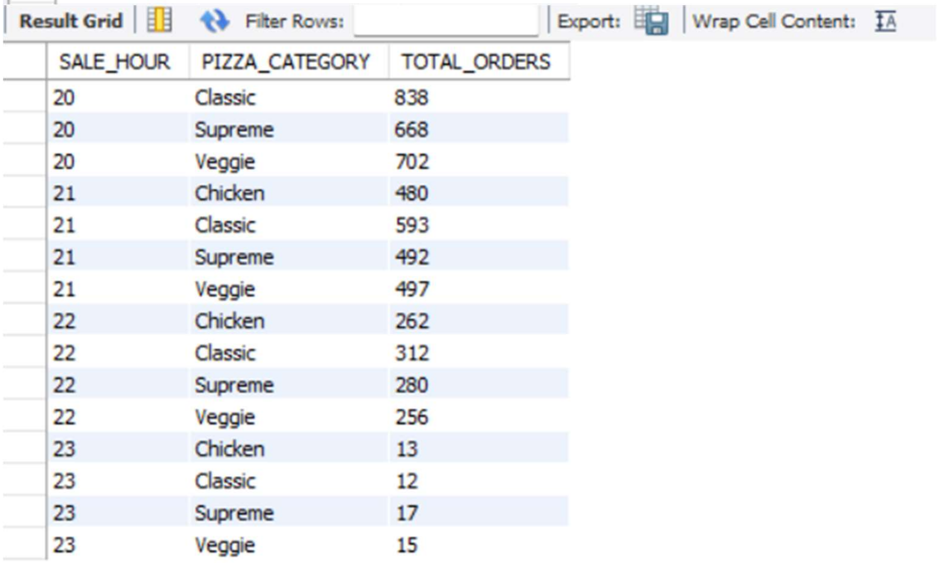
SALE_HOUR	PIZZA_CATEGORY	TOTAL_ORDERS
9	Classic	1
9	Supreme	1
9	Veggie	1
10	Chicken	5
10	Classic	5
10	Supreme	3
10	Veggie	1
11	Chicken	464
11	Classic	613
11	Supreme	500
11	Veggie	460
12	Chicken	1028
12	Classic	1288
12	Supreme	1064
12	Veggie	1067



SALE_HOUR	PIZZA_CATEGORY	TOTAL_ORDERS
13	Chicken	950
13	Classic	1232
13	Supreme	1005
13	Veggie	1021
14	Chicken	577
14	Classic	752
14	Supreme	613
14	Veggie	616
15	Chicken	585
15	Classic	719
15	Supreme	638
15	Veggie	610
16	Chicken	760
16	Classic	999
16	Supreme	839



SALE_HOUR	PIZZA_CATEGORY	TOTAL_ORDERS
16	Veggie	809
17	Chicken	965
17	Classic	1245
17	Supreme	1006
17	Veggie	989
18	Chicken	981
18	Classic	1247
18	Supreme	1066
18	Veggie	034
19	Chicken	816
19	Classic	1003
19	Supreme	893
19	Veggie	863
20	Chicken	650
20	Classic	838

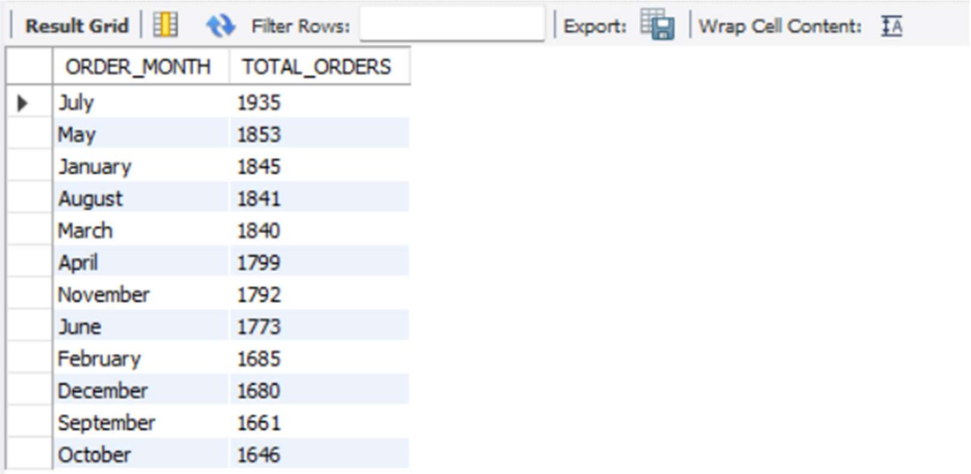


SALE_HOUR	PIZZA_CATEGORY	TOTAL_ORDERS
20	Classic	838
20	Supreme	668
20	Veggie	702
21	Chicken	480
21	Classic	593
21	Supreme	492
21	Veggie	497
22	Chicken	262
22	Classic	312
22	Supreme	280
22	Veggie	256
23	Chicken	13
23	Classic	12
23	Supreme	17
23	Veggie	15

3. MONTHLY TRENDS FOR TOTAL ORDERS

Create a line chart that illustrates the monthly trend of total orders throughout the month. This chart will allow us to identify peak months of high order activity.

```
SELECT MONTHNAME(ORDER_DATE) AS ORDER_MONTH,
COUNT (DISTINCT ORDER_ID) AS TOTAL_ORDERS
FROM PIZZA_SALES
GROUP BY MONTHNAME(ORDER_DATE)
ORDER BY TOTAL_ORDERS DESC;
```

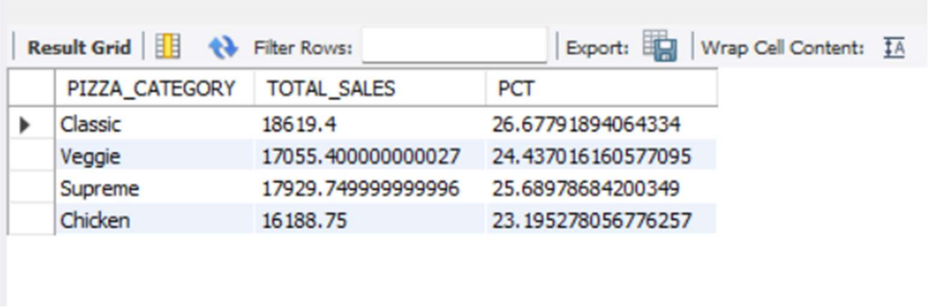


ORDER_MONTH	TOTAL_ORDERS
July	1935
May	1853
January	1845
August	1841
March	1840
April	1799
November	1792
June	1773
February	1685
December	1680
September	1661
October	1646

4. 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.

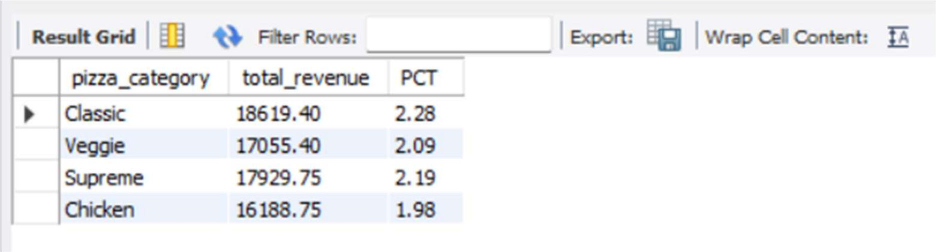
```
SELECT PIZZA_CATEGORY , SUM(TOTAL_PRICE) AS
TOTAL_SALES,
SUM (TOTAL_PRICE)*100 /( SELECT SUM(TOTAL_PRICE) FROM
PIZZA_SALES AS PCT
FROM PIZZA_SALES
GROUP BY PIZZA_CATEGORY;
```



PIZZA_CATEGORY	TOTAL_SALES	PCT
Classic	18619.4	26.6779189406434
Veggie	17055.400000000027	24.4370161605770 95
Supreme	17929.749999999996	25.68978684200349
Chicken	16188.75	23.195278056776257

5. We can calculate it by this method. Here we find percentage of sales by pizza category for January month

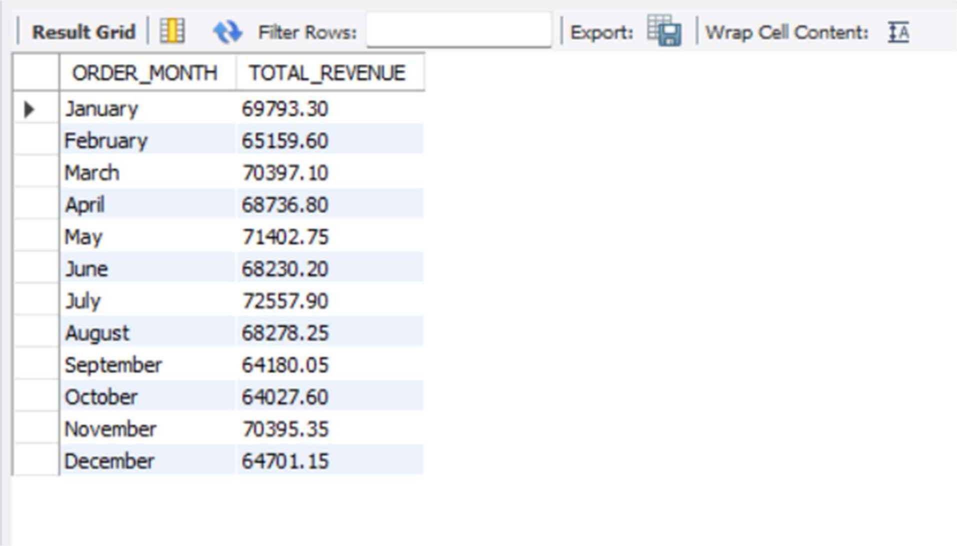
```
SELECT PIZZA_CATEGORY, CAST(SUM(TOTAL_PRICE) AS
DECIMAL (10,2)) AS TOTAL_REVENUE,
CAST(SUM(TOTAL_PRICE) * 100 / (SELECT SUM(TOTAL_PRICE)
FROM PIZZA_SALES) AS DECIMAL (10,2)) AS PCT
FROM PIZZA_SALES
WHERE MONTH (ORDER_DATE )=1
GROUP BY PIZZA_CATEGORY;
```



pizza_category	total_revenue	PCT
Classic	18619.40	2.28
Veggie	17055.40	2.09
Supreme	17929.75	2.19
Chicken	16188.75	1.98

6. TOTAL REVENUE BY MONTH

```
SELECT MONTHNAME(ORDER_DATE) AS ORDER_MONTH,
CAST(SUM(TOTAL_PRICE) AS DECIMAL (10,2)) AS
TOTAL_REVENUE
FROM PIZZA_SALES
GROUP BY MONTHNAME(ORDER_DATE)
ORDER BY TOTAL_REVENUE DESC;
```

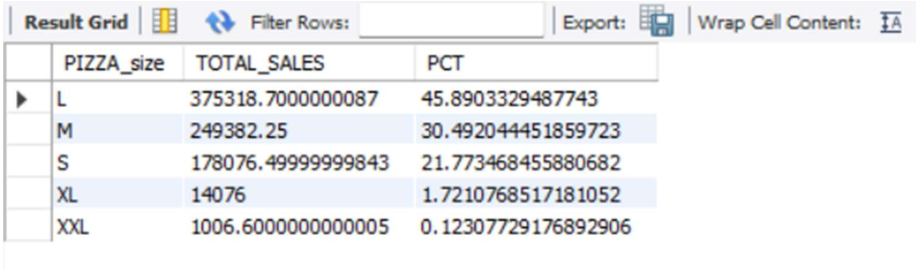


ORDER_MONTH	TOTAL_REVENUE
January	69793.30
February	65159.60
March	70397.10
April	68736.80
May	71402.75
June	68230.20
July	72557.90
August	68278.25
September	64180.05
October	64027.60
November	70395.35
December	64701.15

7. Percentage of sales by pizza size

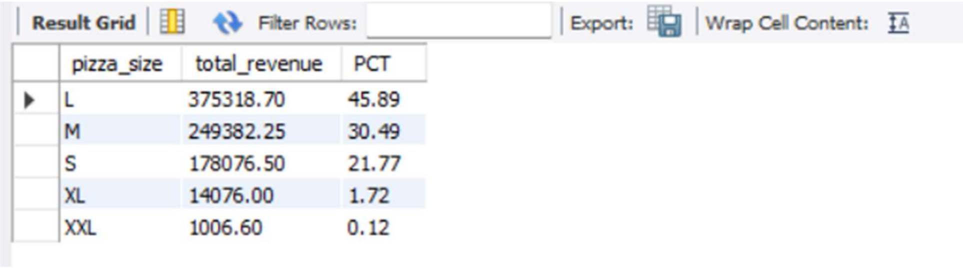
Generate a pie chart that represents the percentage of sales attributed to different pizza size. This chart will help us understand customer preferences for pizza sizes and their impact on sales.

```
SELECT PIZZA_SIZE, SUM(TOTAL_PRICE) AS TOTAL_SALES, SUM
(TOTAL_PRICE) *100 /( SELECT SUM(TOTAL_PRICE) FROM
PIZZA_SALES) AS PCT
FROM PIZZA_SALES
GROUP BY PIZZA_SIZE
ORDER_BY PCT DESC;
```



	PIZZA_size	TOTAL_SALES	PCT
▶	L	375318.7000000087	45.8903329487743
	M	249382.25	30.492044451859723
	S	178076.49999999843	21.773468455880682
	XL	14076	1.7210768517181052
	XXL	1006.6000000000005	0.12307729176892906

```
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_SIZE) AS DECIMAL (10,2)) AS PCT
FROM PIZZA_SALES
GROUP BY PIZZA_SIZE
ORDER_BY PCT DESC;
```

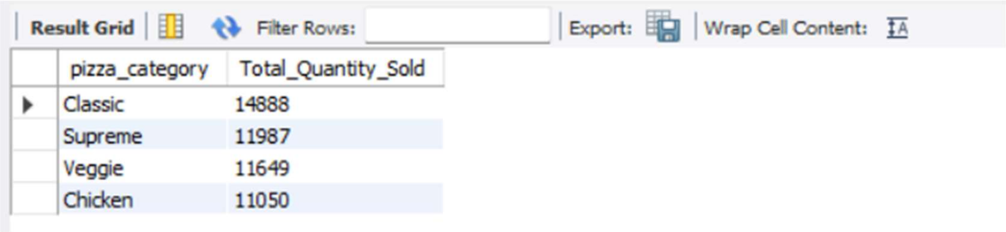


	pizza_size	total_revenue	PCT
▶	L	375318.70	45.89
	M	249382.25	30.49
	S	178076.50	21.77
	XL	14076.00	1.72
	XXL	1006.60	0.12

8. Total pizza sold by pizza category

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

```
SELECT PIZZA_CATEGORY, SUM(QUANTITY) AS
TOTAL_QUANTITY_SOLD
FROM PIZZA_SALES
GROUP BY PIZZA_CATEGORY
ORDER BY TOTAL_QUANTITY_SOLD DESC
```

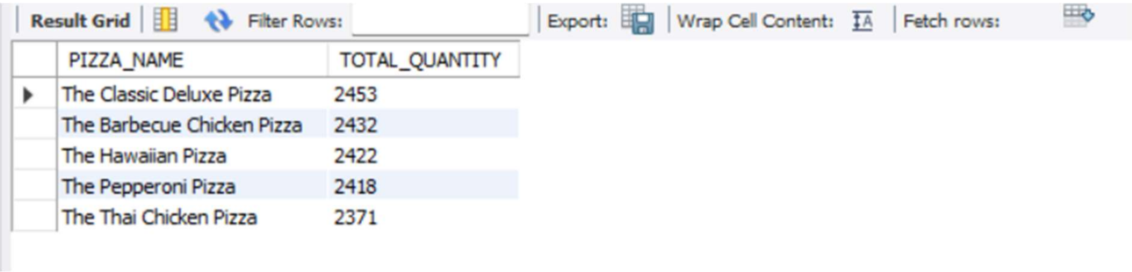


pizza_category	Total_Quantity_Sold
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

9. Top 5 best sellers by total pizza sold

create a bar chart highlighting the top -5 bestselling pizzas based on the total number of pizzas sold. This chart will help us to identify the most popular pizza options.

```
SELECT PIZZA_NAME, SUM(QUANTITY) AS TOTAL_QUANTITY
FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_QUANTITY DESC
LIMIT 5;
```

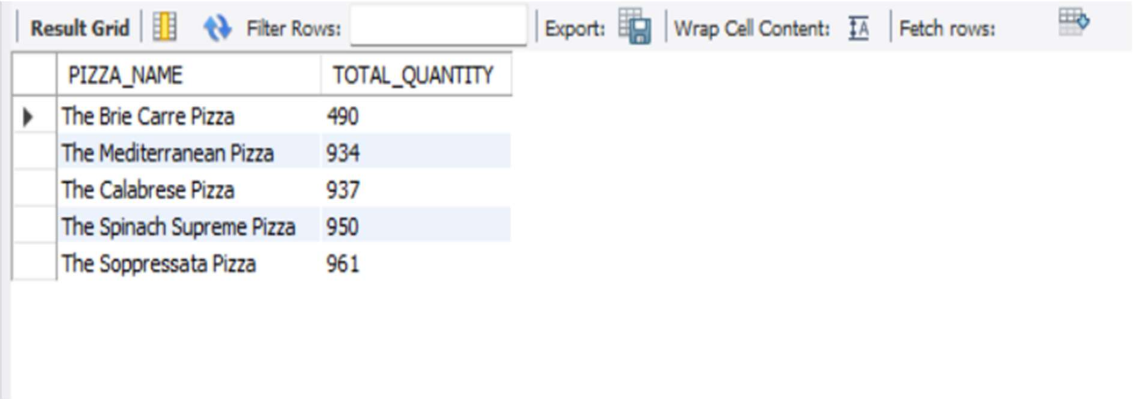


PIZZA_NAME	TOTAL_QUANTITY
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

10.Bottom 5 rows sellers by total pizzas sold

create a bar chart showcasing the bottom 5 worst selling pizzas based on the total number of pizzas sold. This chart enables us to identify underperforming or less popular pizza options.

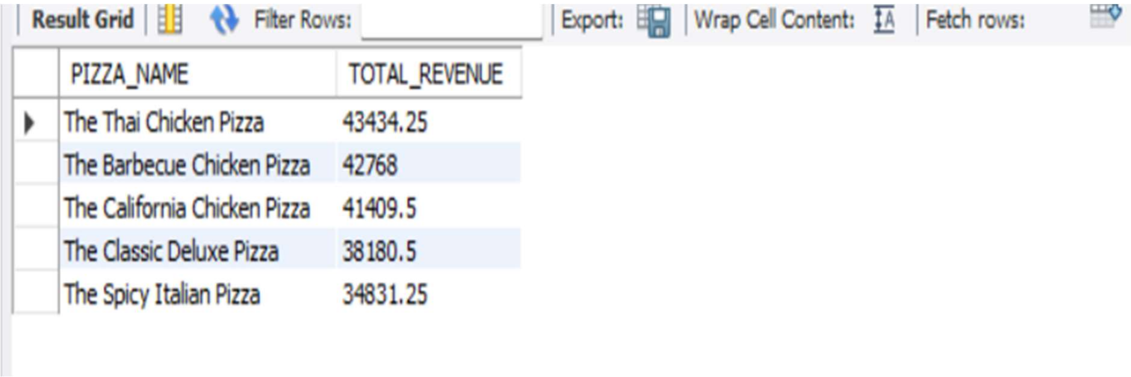
```
SELECT PIZZA_NAME, SUM(QUANTITY) AS TOTAL_QUANTITY
FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_QUANTITY ASC
LIMIT 5;
```



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

11.TOP 5 PIZZA BY REVENUE

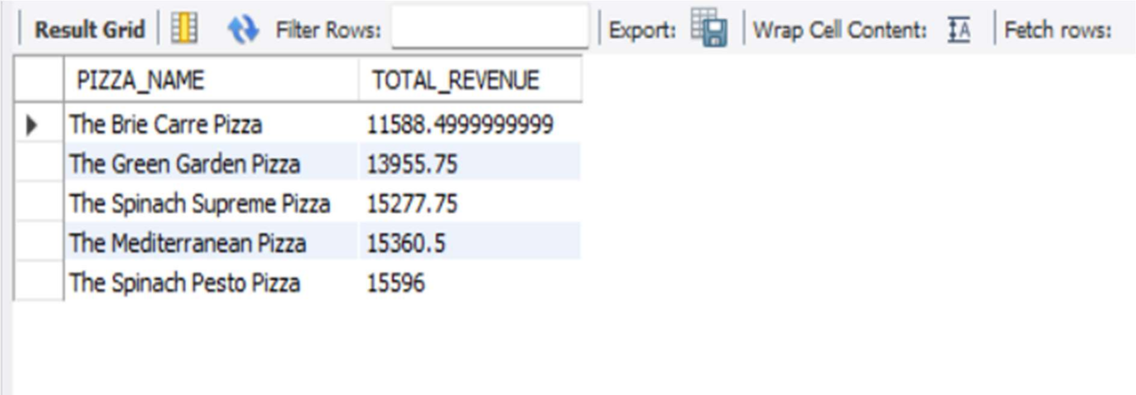
```
SELECT PIZZA_NAME, SUM(TOTAL_PRICE) AS
TOTAL_REVENUE FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_REVENUE DESC
LIMIT 5;
```



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

12.BOTTOM 5 PIZZAS BY REVENUE

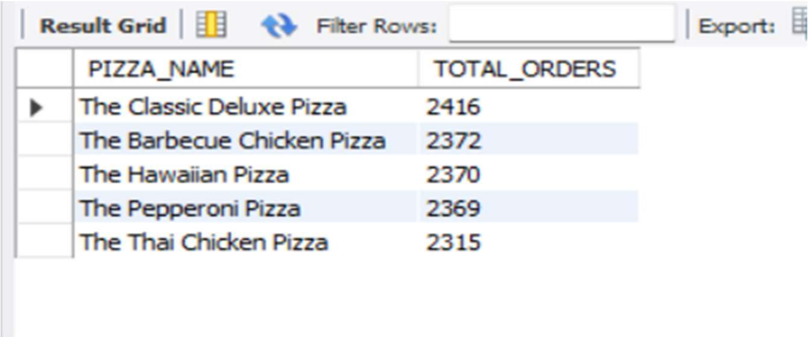
```
SELECT PIZZA_NAME, SUM(TOTAL_PRICE) AS
TOTAL_REVENUE FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_REVENUE ASC
LIMIT 5;
```



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

13.TOP 5 PIZZAS BY TOTAL ORDERS

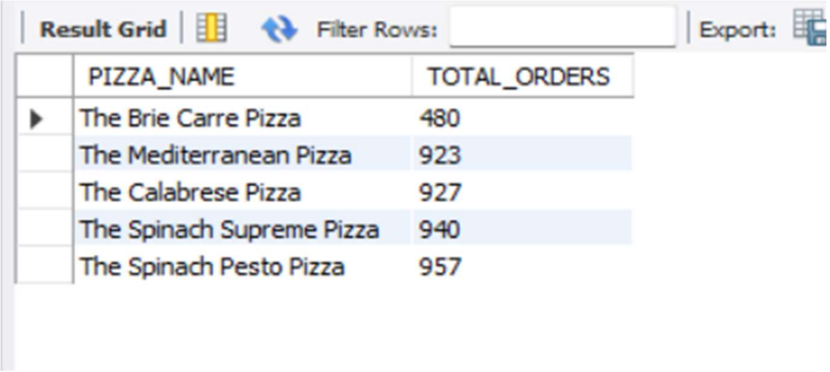
```
SELECT PIZZA_NAME, COUNT(ORDER_ID) AS
TOTAL_ORDERS FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_ORDERS DESC
LIMIT 5;
```



PIZZA_NAME	TOTAL_ORDERS
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

14.BOTTOM 5 PIZZAS BY TOTAL ORDERS

```
SELECT PIZZA_NAME, COUNT(ORDER_ID) AS
TOTAL_ORDERS FROM PIZZA_SALES
GROUP BY PIZZA_NAME
ORDER BY TOTAL_ORDERS ASC
LIMIT 5;
```



PIZZA_NAME	TOTAL_ORDERS
The Brie Carre Pizza	480
The Mediterranean Pizza	923
The Calabrese Pizza	927
The Spinach Supreme Pizza	940
The Spinach Pesto Pizza	957