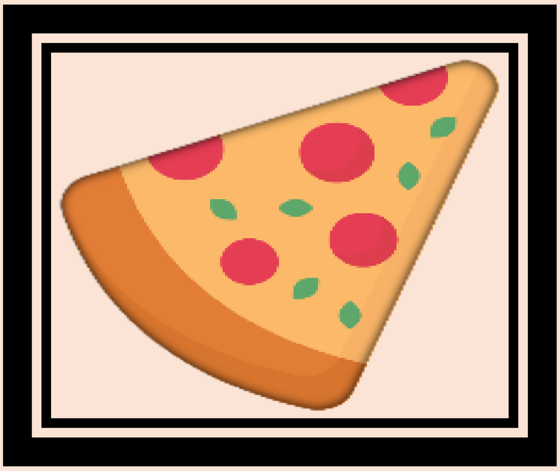


CAPSTONE PROJECT

BY GAURAV VASHIST/PGA 39

MAVEN'S RESTAURANT
PIZZA SALES ANALYSIS

INTRODUCTION

This project analyses pizza sales data to provide valuable insights into product performance, customer preferences, and regional trends. By querying and aggregating relevant information, the project aims to enhance decision-making processes for optimizing inventory, marketing strategies, and overall business efficiency within the pizza industry.

This data set has been imported from the Kaggle, which contains the following information.

Dataset name: pizza distribution restaurant sales. Click here.

This pizza sales dataset makes up 12 relevant features:

- order_id: Unique identifier for each order placed by a table
- order_details_id: Unique identifier for each pizza placed within each order (pizzas of the same type and size are kept in the same row, and the quantity increases)
- pizza_id: Unique key identifier that ties the pizza ordered to its details, like size and price
- quantity: Quantity ordered for each pizza of the same type and size
- order_date: Date the order was placed (entered into the system prior to cooking & serving)
- order_time: Time the order was placed (entered into the system prior to cooking & serving)
- unit_price: Price of the pizza in USD
- total_price: unit_price * quantity
- pizza_size: Size of the pizza (Small, Medium, Large, X Large, or XX Large)
- pizza_type: Unique key identifier that ties the pizza ordered to its details, like size and price
- pizza_ingredients: ingredients used in the pizza as shown in the menu (they all include Mozzarella Cheese, even if not specified; and they all include Tomato Sauce, unless another sauce is specified)
- pizza_name: Name of the pizza as shown in the menu

NOTE

The public dataset is completely available on the Maven Analytics website platform where it stores and consolidates all available datasets for analysis in the Data Playground. **The specific individual datasets at hand can be obtained at this link below:**

https://www.mavenanalytics.io/blog/maven-pizza-challenge.

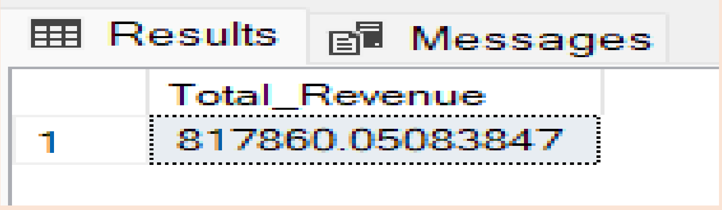
FIRST PART IS TO USE THE MSSQL TO DO THE ANALYSIS.

All the queries are attached with this link. Click here

KPI’s

1. Total Revenue:

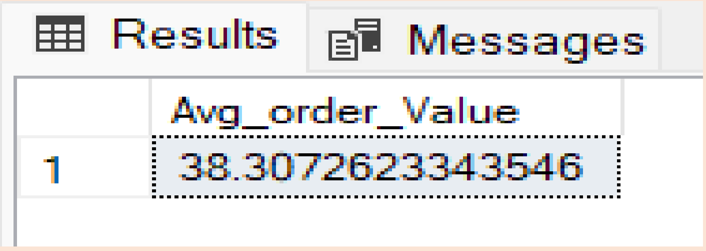
```
SELECT SUM (total_price) AS Total_Revenue FROM pizza_sales;
```



	Total_Revenue
1	817860.05083847

2. Average Order Value

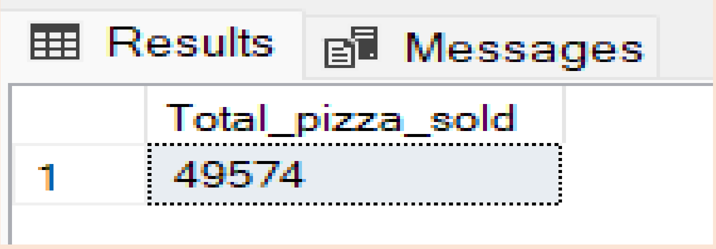
```
SELECT (SUM (total_price) / COUNT (DISTINCT order_id)) AS Avg_order_Value FROM pizza_sales
```



	Avg_order_Value
1	38.3072623343546

3. Total Pizzas Sold

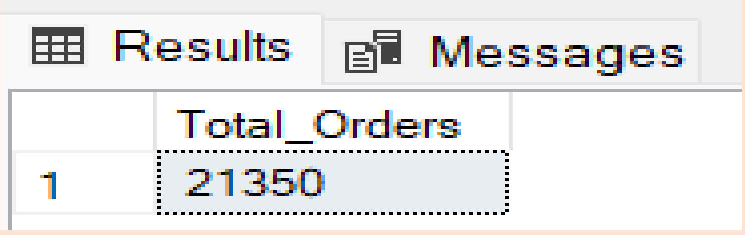
```
SELECT SUM (quantity) AS Total_pizza_sold FROM pizza_sales
```



	Total_pizza_sold
1	49574

4. Total Orders

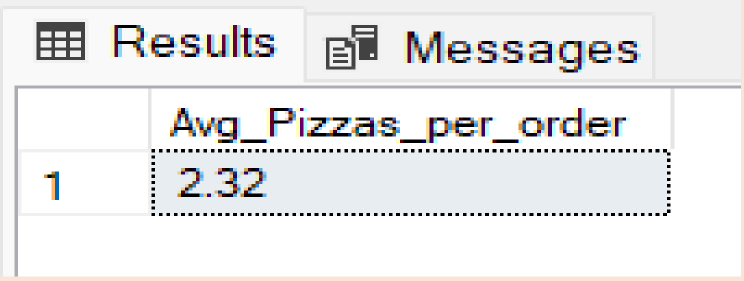
```
SELECT COUNT (DISTINCT order_id) AS Total_Orders FROM pizza_sales
```



	Total_Orders
1	21350

5. Average Pizzas Per Order

```sql
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
```

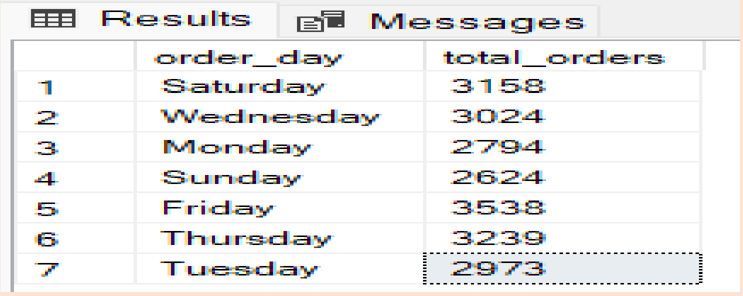


	Avg_Pizzas_per_order
1	2.32

B. Daily Trend for Total Orders

```sql
SELECT DATENAME (DW, order_date) AS order_day, COUNT (DISTINCT order_id) AS
total_orders
FROM pizza_sales
GROUP BY DATENAME (DW, order_date)
```

Output:

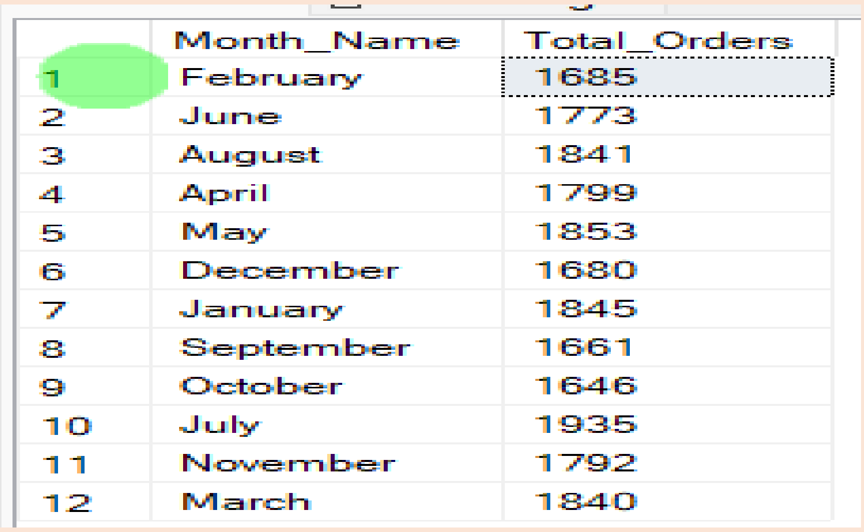


	order_day	total_orders
1	Saturday	3158
2	Wednesday	3024
3	Monday	2794
4	Sunday	2624
5	Friday	3538
6	Thursday	3239
7	Tuesday	2973

C. Monthly Trend for Orders

```sql
select DATENAME (MONTH, order_date) as Month_Name, COUNT (DISTINCT order_id) as
Total_Orders
from pizza_sales
GROUP BY DATENAME (MONTH, order_date)
```

Output

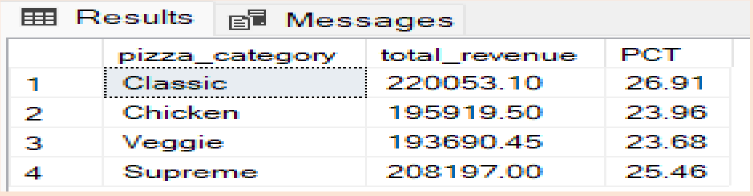


	Month_Name	Total_Orders
1	February	1685
2	June	1773
3	August	1841
4	April	1799
5	May	1853
6	December	1680
7	January	1845
8	September	1661
9	October	1646
10	July	1935
11	November	1792
12	March	1840

D. % of Sales by Pizza Category

```
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
GROUP BY pizza_category
```

Output

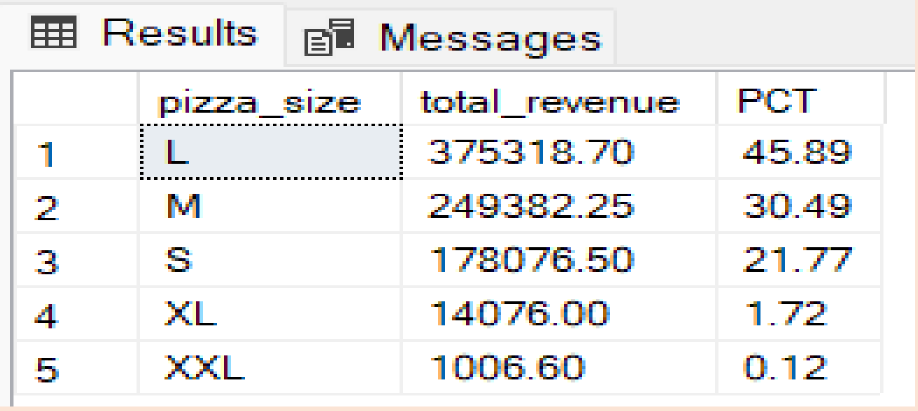


	pizza_category	total_revenue	PCT
1	Classic	220053.10	26.91
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68
4	Supreme	208197.00	25.46

E. % of Sales by Pizza Size

```
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_sales) AS DECIMAL
(10,2)) AS PCT
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size
```

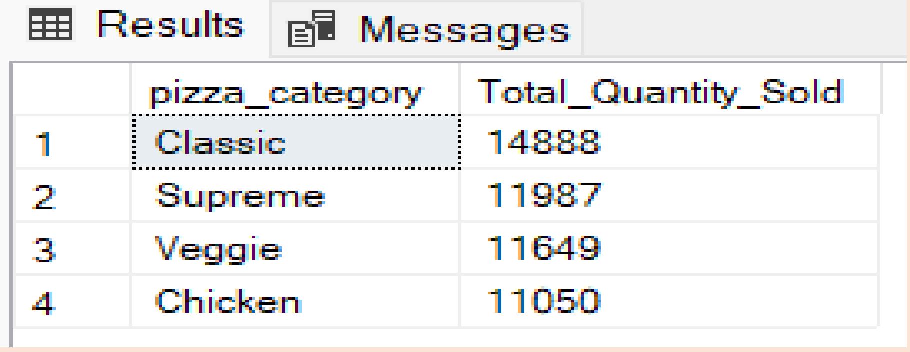
Output



	pizza_size	total_revenue	PCT
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

F. Total Pizzas Sold by Pizza Category

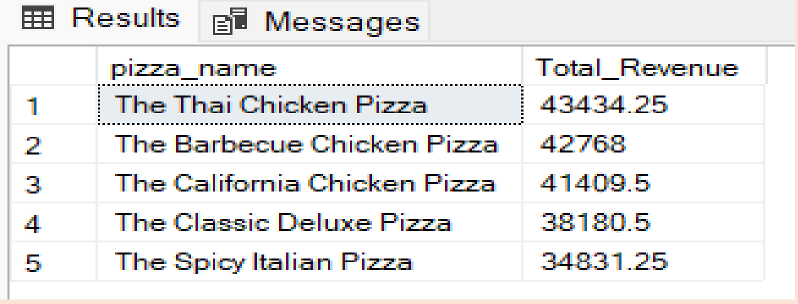
```
SELECT pizza_category, SUM (quantity) as Total_Quantity_Sold
FROM pizza_sales
WHERE MONTH (order_date) = 2
GROUP BY pizza_category
ORDER BY Total_Quantity_Sold DESC
```



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

G. Top 5 Pizzas by Revenue

```
SELECT Top 5 pizza_name, SUM (total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue DESC
```

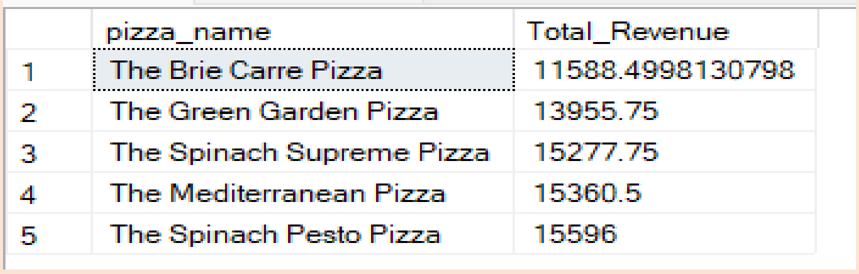


	pizza_name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Spicy Italian Pizza	34831.25

H. Bottom 5 Pizzas by Revenue

```
SELECT Top 5 pizza_name, SUM (total_price) AS Total_Revenue
```

```
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC
```

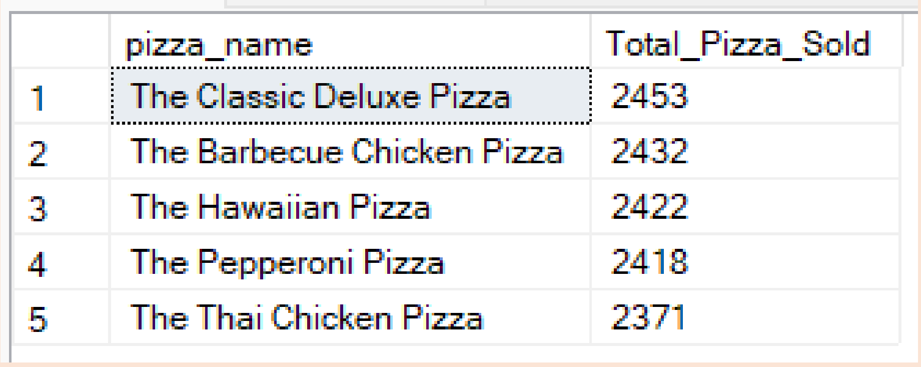


	pizza_name	Total_Revenue
1	The Brie Carre Pizza	11588.4998130798
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.5
5	The Spinach Pesto Pizza	15596

I. Top 5 Pizzas by Quantity

```
SELECT Top 5 pizza_name, SUM (quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold DESC
```

Output



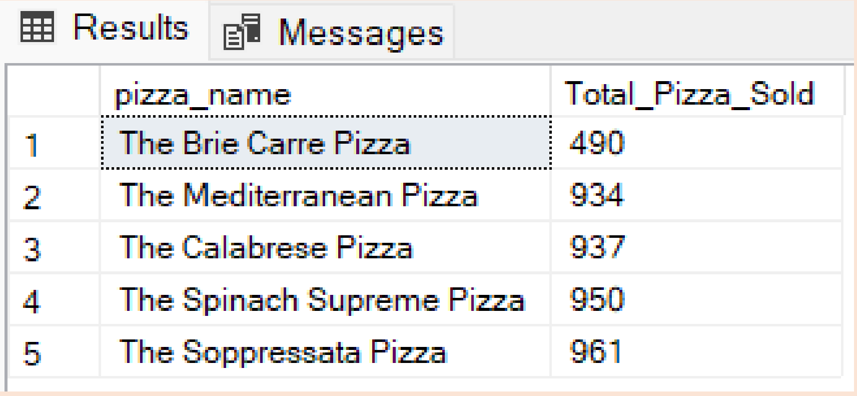
	pizza_name	Total_Pizza_Sold
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

J. Bottom 5 Pizzas by Quantity

```
SELECT TOP 5 pizza_name, SUM (quantity) AS Total_Pizza_Sold

FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold ASC
```

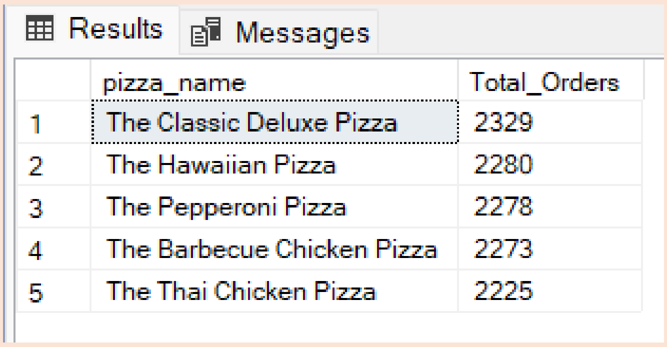
Output



	pizza_name	Total_Pizza_Sold
1	The Brie Carre Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961

K. Top 5 Pizzas by Total Orders

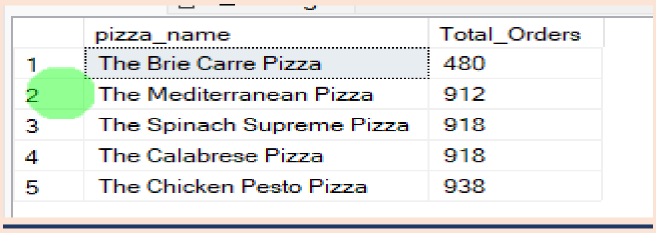
```
SELECT Top 5 pizza_name, COUNT (DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders DESC
```



	pizza_name	Total_Orders
1	The Classic Deluxe Pizza	2329
2	The Hawaiian Pizza	2280
3	The Pepperoni Pizza	2278
4	The Barbecue Chicken Pizza	2273
5	The Thai Chicken Pizza	2225

L. Bottom 5 Pizzas by Total Orders

```
SELECT Top 5 pizza_name, COUNT (DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders ASC
```



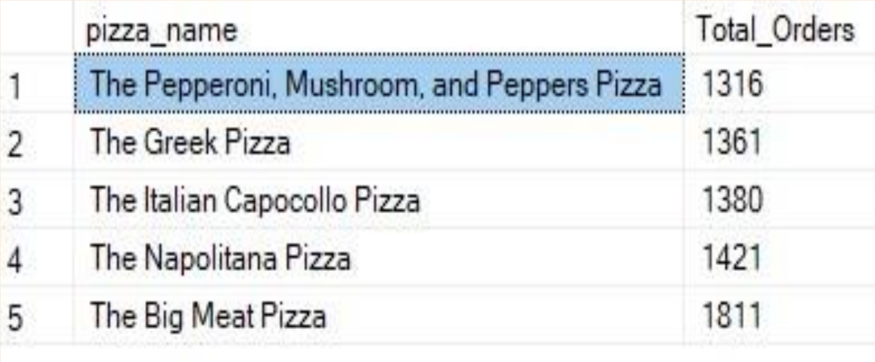
	pizza_name	Total_Orders
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Spinach Supreme Pizza	918
4	The Calabrese Pizza	918
5	The Chicken Pesto Pizza	938

NOTE

If you want to apply the pizza_category or pizza_size filters to the above queries you can use WHERE clause. Follow some of below examples

```
SELECT Top 5 pizza_name, COUNT (DISTINCT order_id) AS Total_Orders
FROM pizza_sales
WHERE pizza_category = 'Classic'
GROUP BY pizza_name
```

```
ORDER BY Total_Orders ASC
```

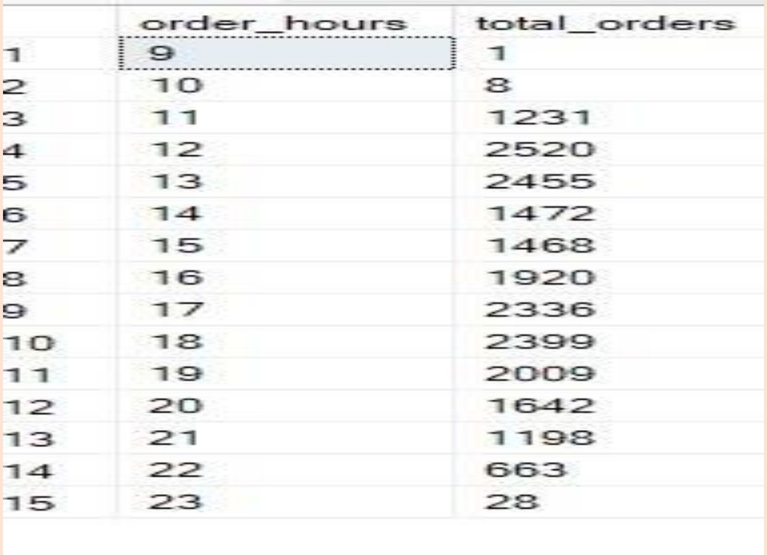


	pizza_name	Total_Orders
1	The Pepperoni, Mushroom, and Peppers Pizza	1316
2	The Greek Pizza	1361
3	The Italian Capocollo Pizza	1380
4	The Napolitana Pizza	1421
5	The Big Meat Pizza	1811

Orders Hours.

-- Hourly Trend

```
 select datepart(hour,order_time) as order_hours,  count(distinct order_id) as
total_orders from
 pizza_sales
 group by datepart(hour,order_time)
 order by datepart(hour,order_time)
```

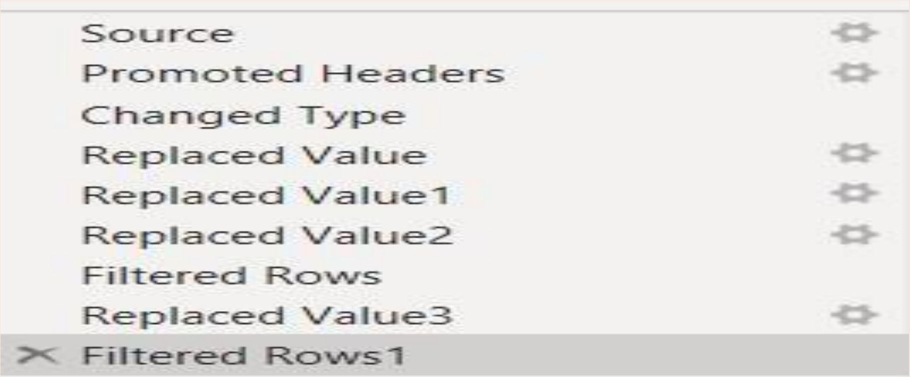


	order_hours	total_orders
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

Analysis of the Project By using the POWER BI

Leveraging Power BI for pizza sales analysis, integrated with MS SQL data, revealed key insights into customer preferences, popular toppings, and sales trends. The interactive dashboards showcased clear visualizations, facilitating informed decision-making for optimizing inventory, marketing strategies, and enhancing overall business performance.

1 Transformation of the Data by using the POWER QUERY TOOL.



Source
Promoted Headers
Changed Type
Replaced Value
Replaced Value1
Replaced Value2
Filtered Rows
Replaced Value3
Filtered Rows1

2 Creation of the new measures by using the DAX queries.

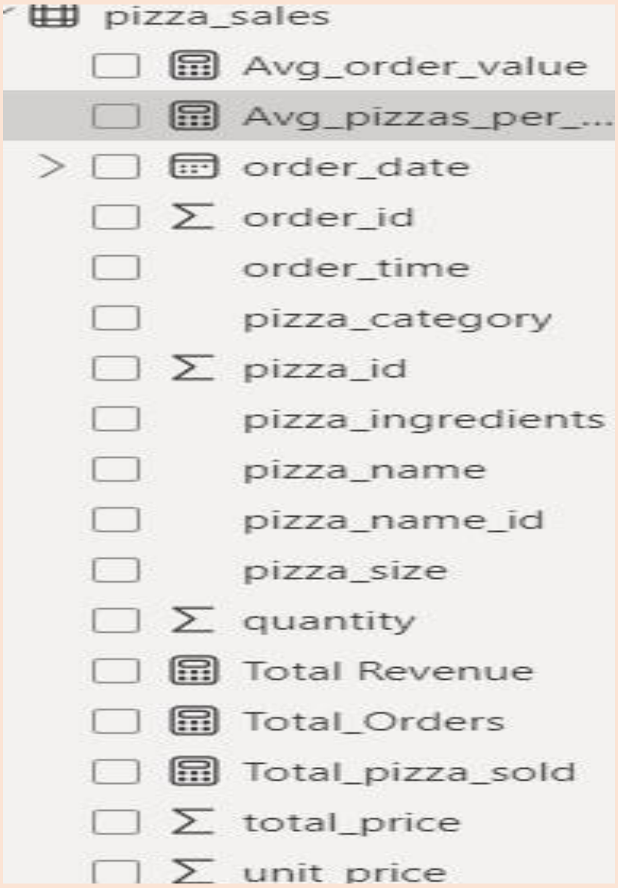
Total Revenue = sum(pizza_sales[total_price])

Total_Orders = DISTINCT (pizza_sales[order_id])

Total_pizza_sold = sum(pizza_sales[quantity])

Avg_order_value = [Total Revenue]/[Total_Orders]

Avg_pizzas_per_order = [Total_pizza_sold]/[Total_Orders]



pizza_sales

□ Avg_order_value

□ Avg_pizzas_per_...

□ order_date

□ order_id

□ order_time

□ pizza_category

□ pizza_id

□ pizza_ingredients

□ pizza_name

□ pizza_name_id

□ pizza_size

□ quantity

□ Total Revenue

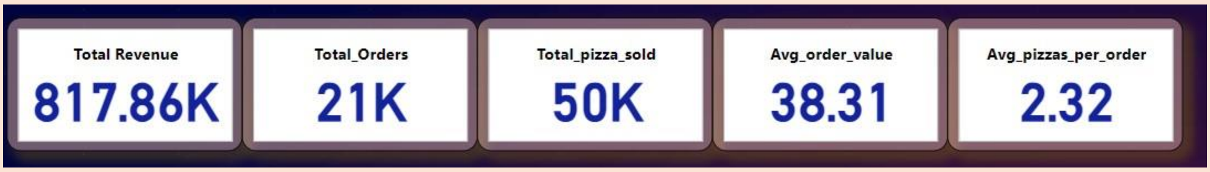
□ Total_Orders

□ Total_pizza_sold

□ total_price

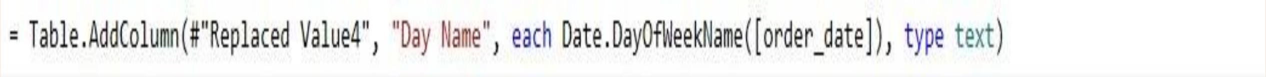
□ unit_price

3 Start creating the KPI’s (Visual Cards).



Total Revenue	Total_Orders	Total_pizza_sold	Avg_order_value	Avg_pizzas_per_order
817.86K	21K	50K	38.31	2.32

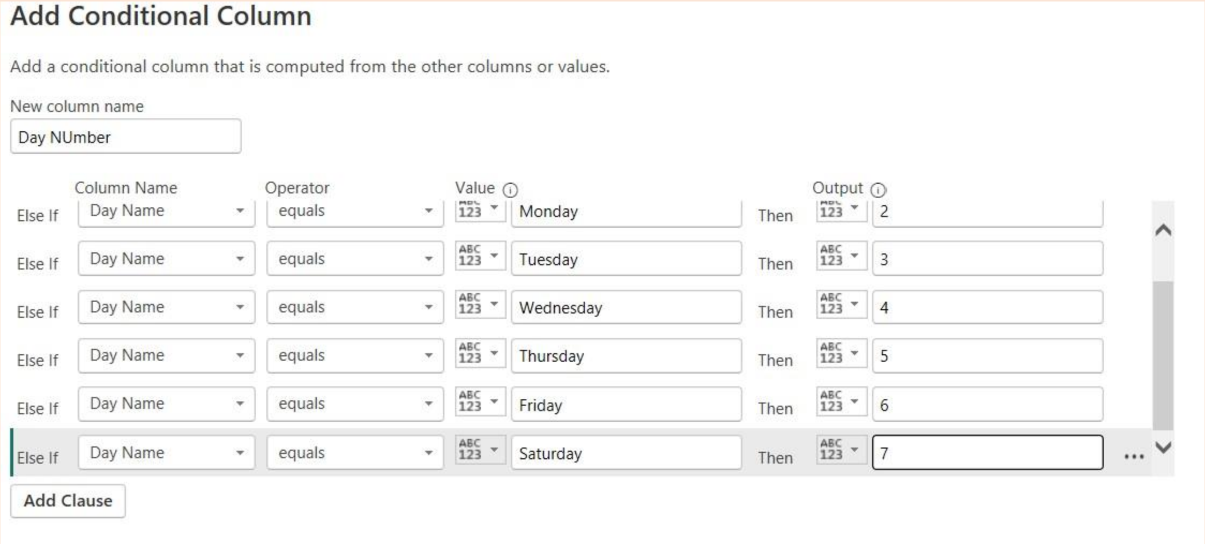
4 To make the chart create a new column Day by using the power query tool.



```
= Table.AddColumn(#"Replaced Value4", "Day Name", each Date.DayOfWeekName([order_date]), type text)
```

Order_Day = UPPER (LEFT (pizza_sales [Day Name],3)): This is used to get first 3 letters from day name in the Capital format.

5 To make a stack column chart week day name in the proper sorting order, then create a Conditional column through the DAX command.



Add Conditional Column

Add a conditional column that is computed from the other columns or values.

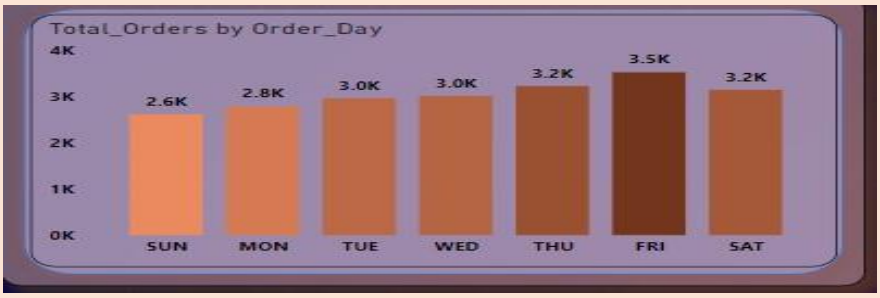
New column name

Day NUmber

	Column Name	Operator	Value		Output
Else If	Day Name	equals	Monday	Then	2
Else If	Day Name	equals	Tuesday	Then	3
Else If	Day Name	equals	Wednesday	Then	4
Else If	Day Name	equals	Thursday	Then	5
Else If	Day Name	equals	Friday	Then	6
Else If	Day Name	equals	Saturday	Then	7

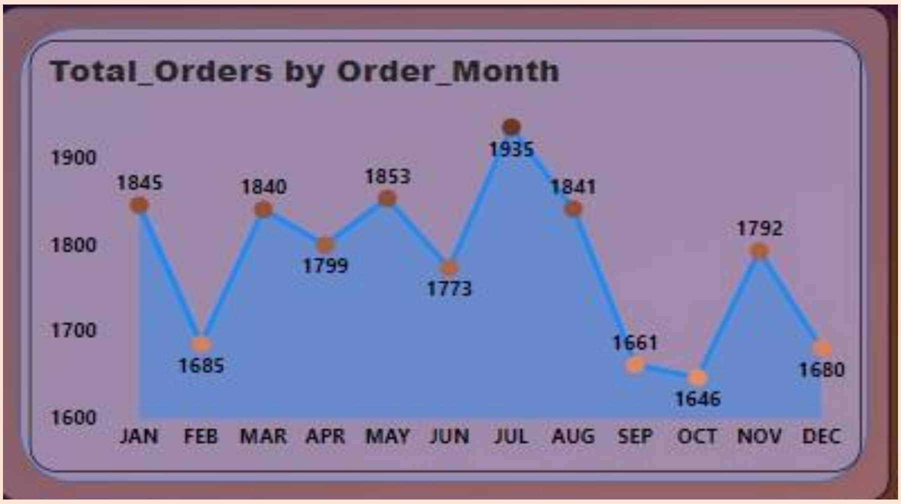
Then use sort column command and with the help of sort axis we can achieve the desirable output.

6 Total order by Day

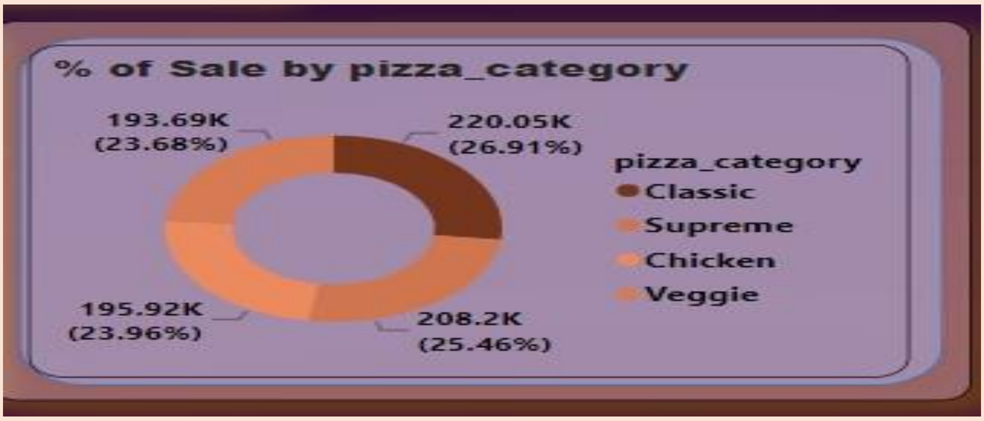


7 Total order by Month.

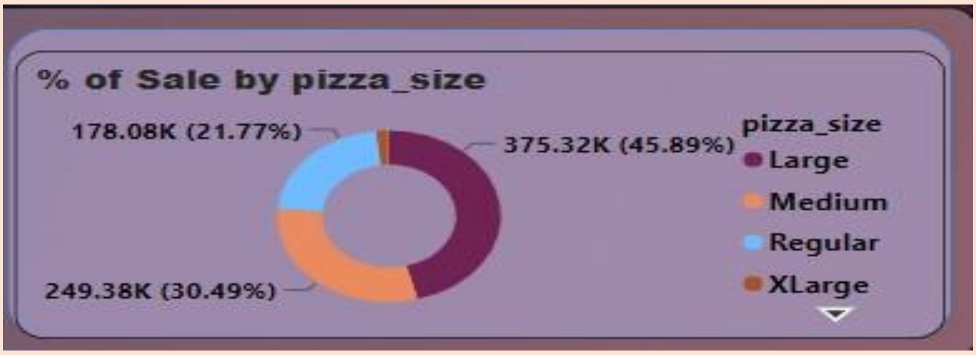
Create a new column by using power query and extract the number of month and month from add column (Date Option).



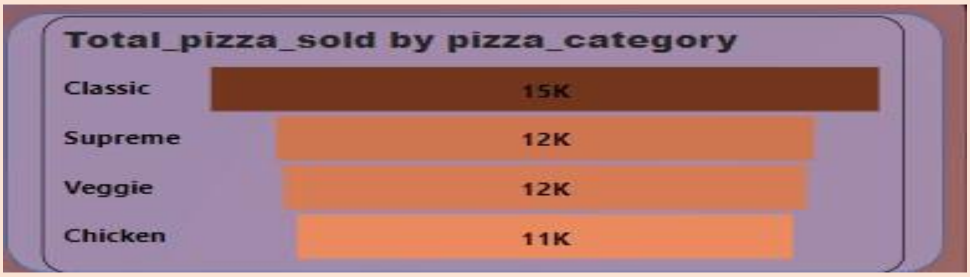
8 % of Sales by Pizza Category.



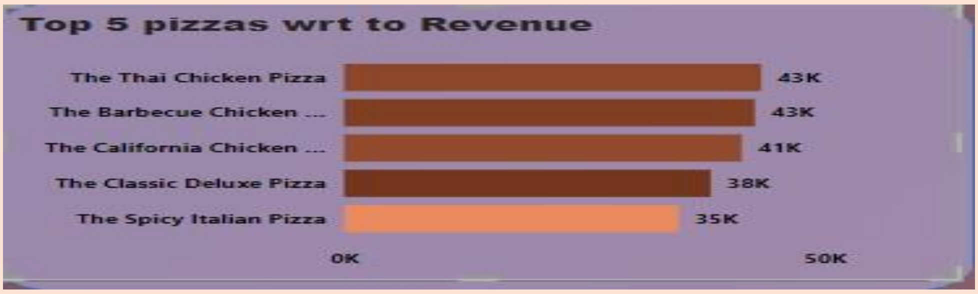
9 % of Sales by Pizza Size.



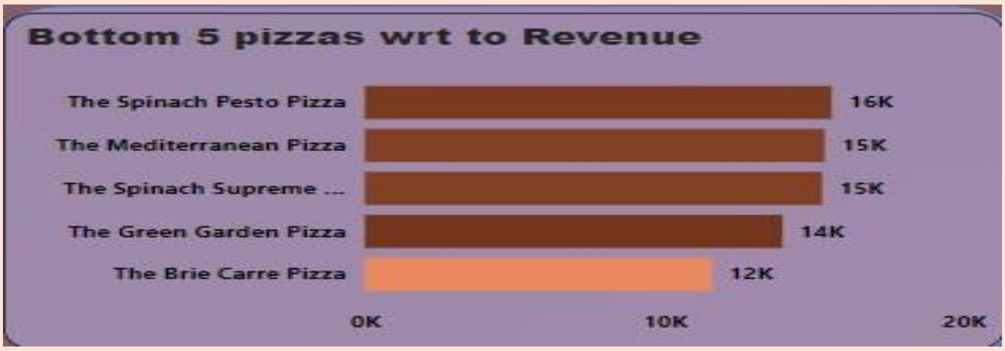
10 Total pizzas sold by Pizza category.



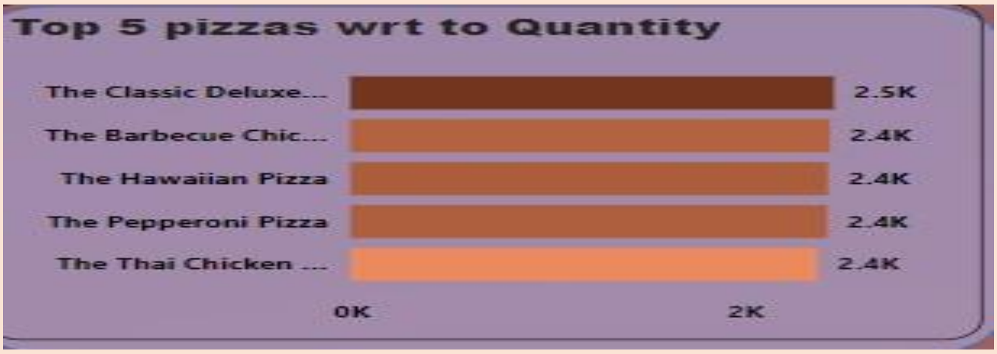
11 Top 5 pizzas w.r.t to Revenue.



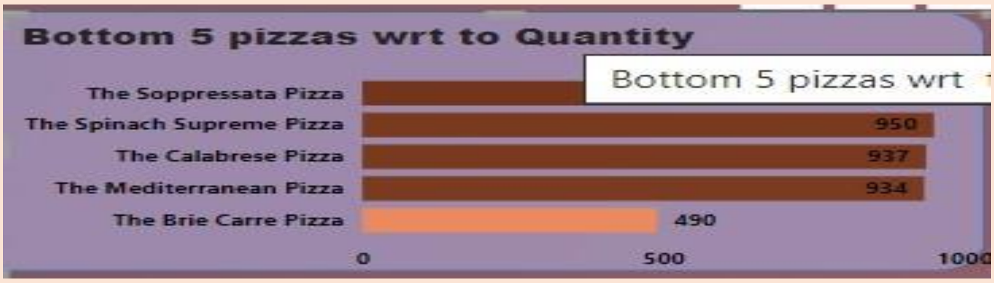
12 Bottom 5 w.r.t the Revenue.



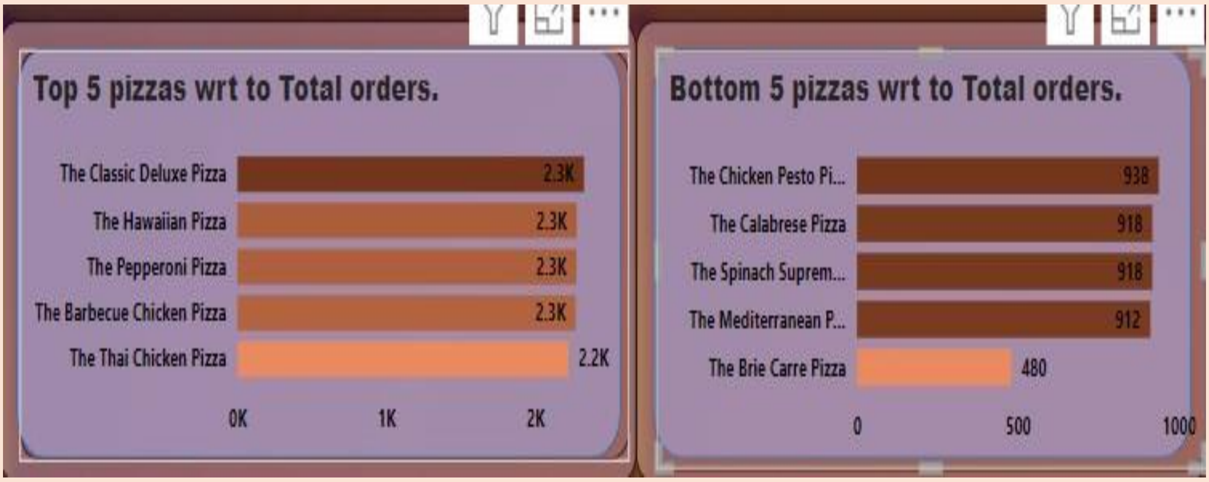
13 Top 5 Pizzas w.r.t Quantity.



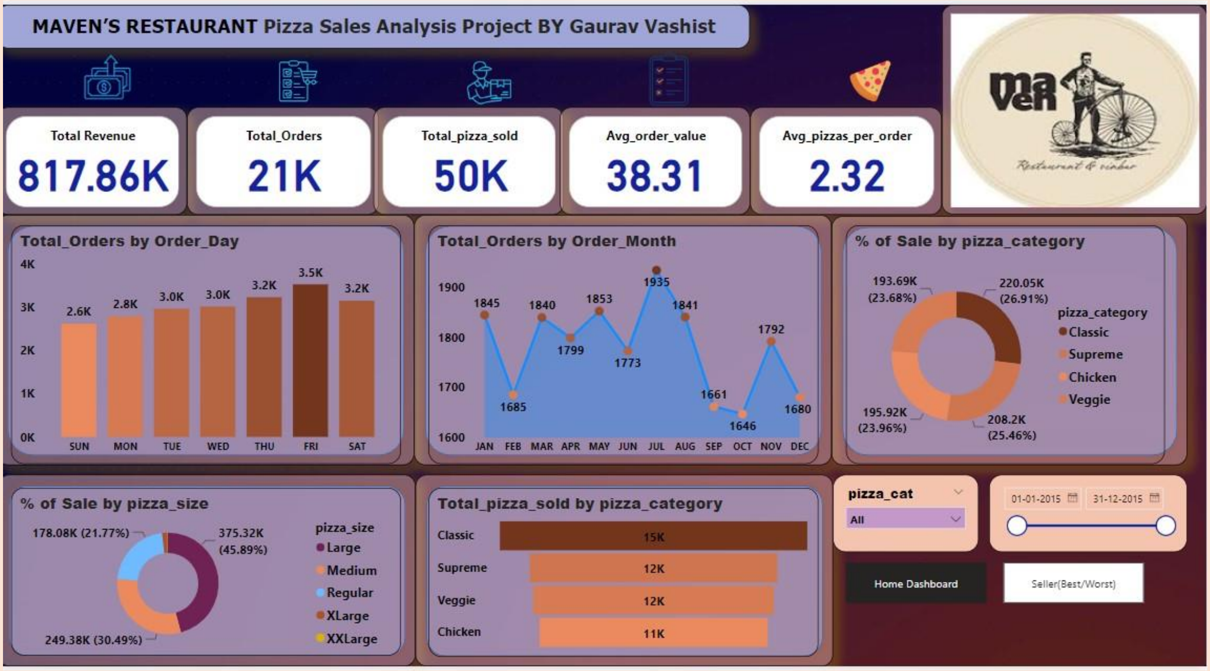
14 Bottom 5 Pizzas Sold.



15. Top and Bottom 5 By Total Orders.



Final Dashboard Analysis


MAVEN'S RESTAURANT Pizza Sales Analysis Project BY Gaurav Vashist

Total Revenue 817.86K

Total_Orders 21K

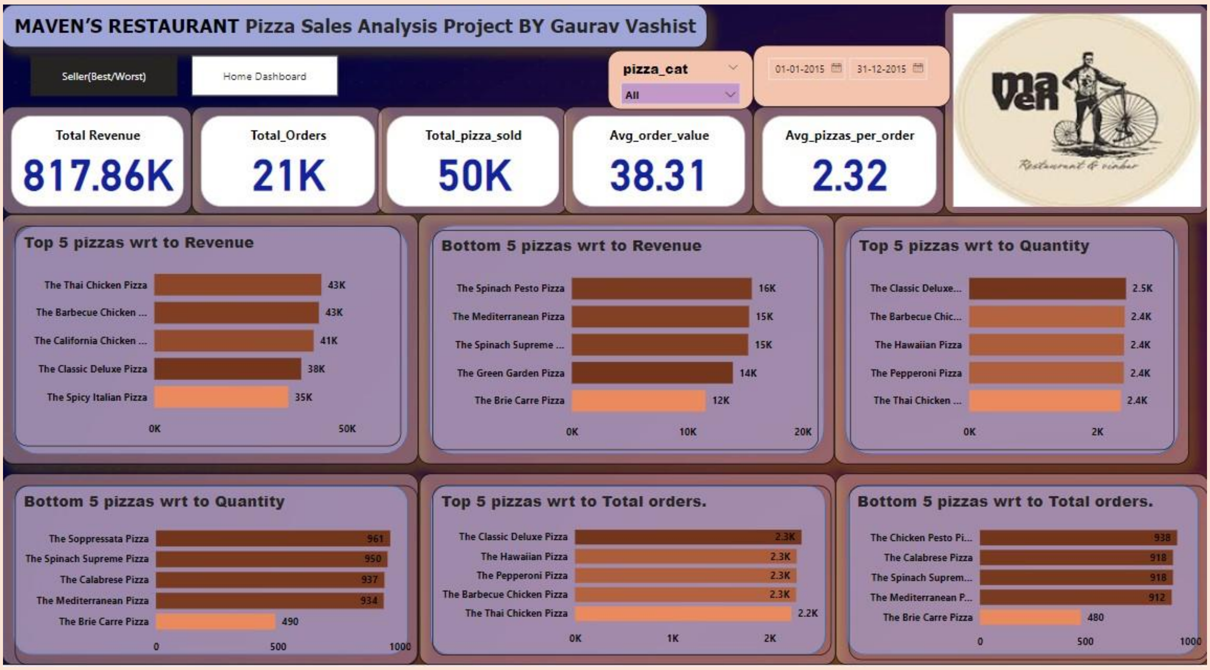
Total_pizza_sold 50K

Avg_order_value 38.31

Avg_pizzas_per_order 2.32

pizza_cat: All

01-01-2015 – 31-12-2015


MAVEN'S RESTAURANT Pizza Sales Analysis Project BY Gaurav Vashist

Total Revenue 817.86K

Total_Orders 21K

Total_pizza_sold 50K

Avg_order_value 38.31

Avg_pizzas_per_order 2.32

pizza_cat: All

01-01-2015 – 31-12-2015

Seller(Best/Worst)
Home Dashboard
maVeR
Restaurant & vinbar
Top 5 pizzas wrt to Revenue
The Thai Chicken Pizza 43K
The Barbecue Chicken ... 43K
The California Chicken ... 41K
The Classic Deluxe Pizza 38K
The Spicy Italian Pizza 35K
0K
50K
Bottom 5 pizzas wrt to Revenue
The Spinach Pesto Pizza 16K
The Mediterranean Pizza 15K
The Spinach Supreme ... 15K
The Green Garden Pizza 14K
The Brie Carre Pizza 12K
0K
10K
20K
Top 5 pizzas wrt to Quantity
The Classic Deluxe... 2.5K
The Barbecue Chic... 2.4K
The Hawaiian Pizza 2.4K
The Pepperoni Pizza 2.4K
The Thai Chicken ... 2.4K
0K
2K
Bottom 5 pizzas wrt to Quantity
The Soppressata Pizza 961
The Spinach Supreme Pizza 950
The Calabrese Pizza 937
The Mediterranean Pizza 934
The Brie Carre Pizza 490
0
500
1000
Top 5 pizzas wrt to Total orders.
The Classic Deluxe Pizza 2.3K
The Hawaiian Pizza 2.3K
The Pepperoni Pizza 2.3K
The Barbecue Chicken Pizza 2.3K
The Thai Chicken Pizza 2.2K
0K
1K
2K
Bottom 5 pizzas wrt to Total orders.
The Chicken Pesto Pi... 938
The Calabrese Pizza 918
The Spinach Suprem... 918
The Mediterranean P... 912
The Brie Carre Pizza 480
0
500
1000

Results

- During daily working hours, we see that sales peak between 12 and 1 in the afternoon and between 16 and 19 in the evening.
- The days when pizza is sold the most are Thursday, Friday and Saturday.

- In terms of sales percentage in the pizza category, the classic and supreme categories are only slightly ahead of the chicken and veggie categories.
- Pizza size is listed as Large, Medium and Small in terms of sales percentage, with X-Large and XX-Large having very low percentages.
- In terms of quantity and sales, classic pizza stands out the most on a category basis.
- The best-selling pizza by revenue is The Thai Chicken Pizza, the worst selling pizza is The Brie Carre Pizza
- The best-selling pizza by quantity is The Thai Classic Deluxe Pizza, the worst selling pizza is The Brie Carre Pizza
- The best-selling pizza by sales is The Thai Classic Deluxe Pizza, the worst selling pizza is The Brie Carre Pizza.
- **Insights.**

- Customers do not prefer much pizza between 9-11 in the morning and 21-23 at night. For this reason, different breakfast options can be added between breakfast hours, and a campaign menu can be prepared between 21-23 in the evening.
- Sales can be increased with a campaign on Sundays and Mondays, when pizza sales are lowest.
- Supreme and Veggie categories are the pizza categories with the worst sales, and X-Large and XX-Large sizes are the least sold pizza sizes. These categories and sizes can be evaluated in campaigns.

By using Pandas we can analyse the same project. Click here.