Project : Pizza Sales Analysis

Table of Contents:

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.
2. Average Order Value: The average amount spent per order, calculated by dividing the total revenue by the total number of orders.
3. Total Pizzas Sold: The sum of the quantities of all pizzas sold.
4. Total Orders: The total number of orders placed.
5. Average Pizzas 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.

CHARTS REQUIREMENT

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. Hourly Trend for Total Pizzas Sold: 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.

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.

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.

4. Percentage of Sales 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.

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.

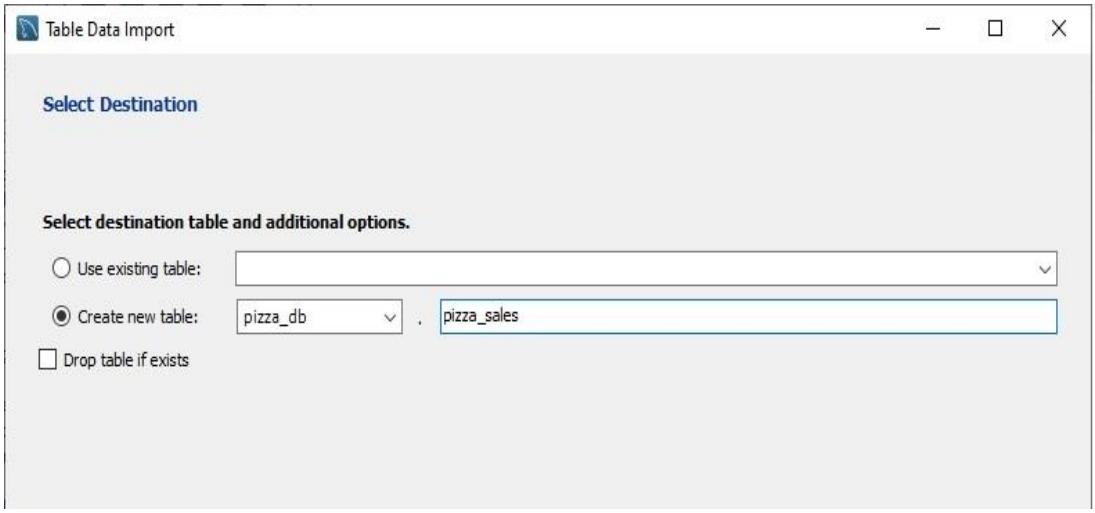
6. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders: Create a bar chart highlighting the top 5 best-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will help us identify the most popular pizza options.

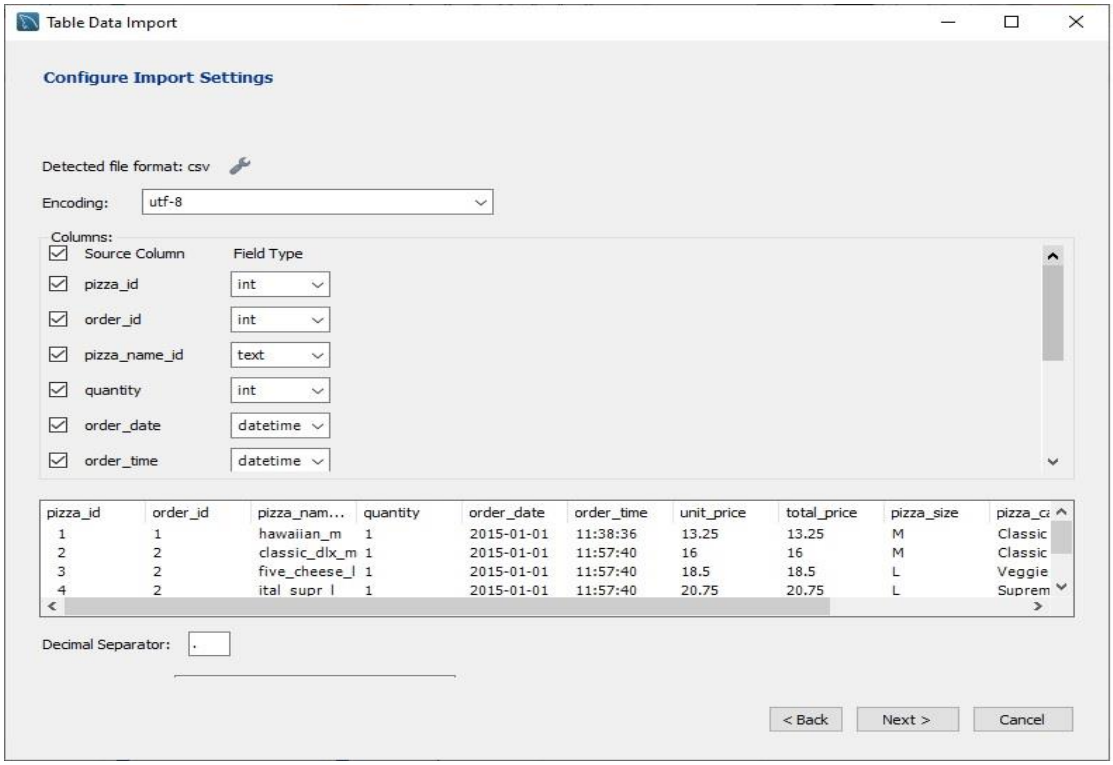
7. Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders: Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will enable us to identify underperforming or less popular pizza options.

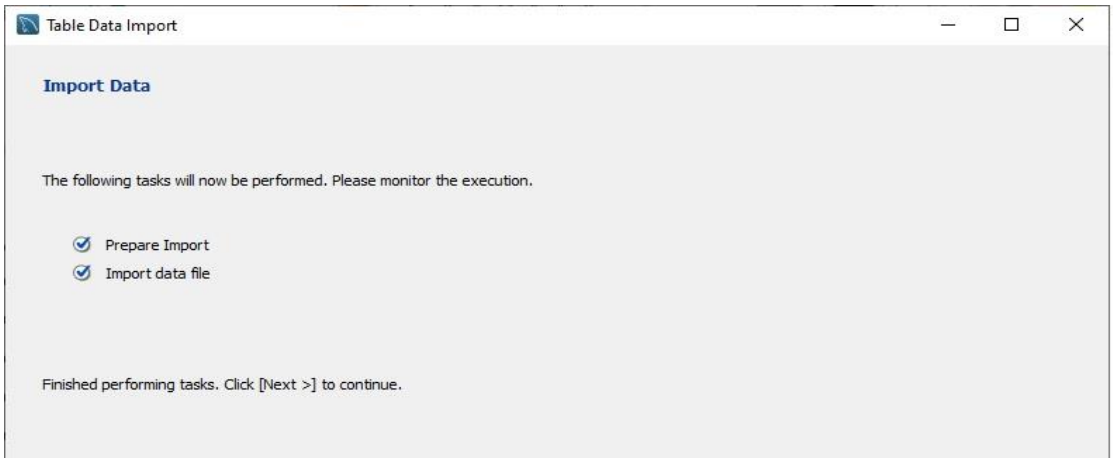
Data Analysis using MySQL

Utilized MySQL for data extraction and calculation of key metrics such as Total Revenue, Average Order Value, Total Pizzas Sold, Total Orders, and Average Pizzas Per Order.

DATA IMPORT





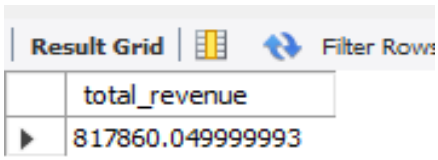


ANALYSIS OF DIFFERENT SQL STATEMENT ON DATA BASE

A. KPI's

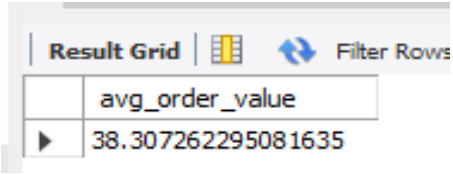
1. Total Revenue:

SELECT SUM(total_price) AS total_revenue FROM pizza_sales;



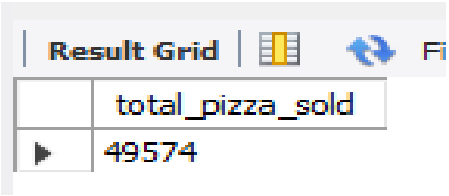
2. Average Order Value:

SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS avg_order_value FROM pizza_sales;



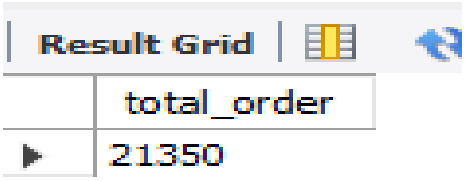
3. Total Pizza Sold

SELECT SUM(quantity) AS total_pizza_sold FROM pizza_sales;



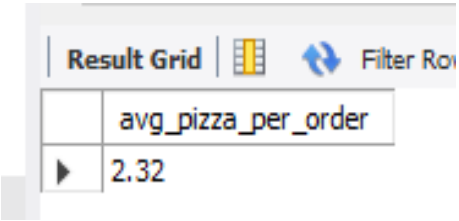
4. Total Orders

SELECT COUNT(DISTINCT order_id) AS total_order FROM pizza_sales;



5. Average Pizzas Per Order

SELECT ROUND(SUM(quantity) / COUNT(DISTINCT order_id), 2) AS avg_pizza_per_order FROM pizza_sales;



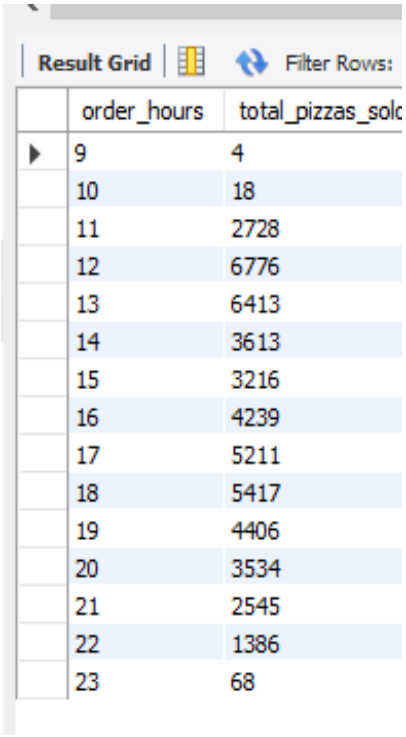
B. Hourly Trend for Total Pizzas Sold

SELECT HOUR(order_time) AS order_hours, SUM(quantity) AS total_pizzas_sold

FROM pizza_sales

GROUP BY HOUR(order_time)

ORDER BY HOUR(order_time);



Result Grid | Filter Rows:

order_hours	total_pizzas_sold
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

C. Weekly Trend for Orders

```
SELECT WEEK(order_date, 3) AS WeekNumber, YEAR(order_date) AS Year,
COUNT(DISTINCT order_id) AS Total_orders
FROM pizza_sales
GROUP BY WEEK(order_date, 3), YEAR(order_date)
ORDER BY Year, WeekNumber;
```

WeekNumber	Year	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

Result Grid | Filter Rows:

WeekNumber	Year	Total_orders
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
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

D. % of Sales by Pizza Category

```
SELECT pizza_category,
    ROUND(SUM(total_price), 2) AS total_revenue,
    ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM pizza_sales), 2) AS PCT
FROM pizza_sales
```

GROUP BY pizza_category;

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

OR

```
SELECT pizza_category,
    ROUND(SUM(total_price), 2) AS total_revenue,
    ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM pizza_sales
     WHERE MONTH(order_date) = 1), 2) AS PCT
 FROM pizza_sales
 WHERE MONTH(order_date) = 1
GROUP BY pizza_category;
```

pizza_category	total_revenue	PCT
Classic	18619.40	26.68
Veggie	17055.40	24.44
Supreme	17929.75	25.69
Chicken	16188.75	23.20

E. % of Sales by Pizza Size

```
SELECT pizza_size,
    ROUND(SUM(total_price), 2) AS total_revenue,
    ROUND(SUM(total_price) * 100 / (SELECT SUM(total_price) FROM pizza_sales), 2) AS
PCT
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size;
```

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

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
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

G. Top 5 Pizzas 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	42768.00
The California Chicken Pizza	41409.50
The Classic Deluxe Pizza	38180.50
The Spicy Italian Pizza	34831.25

H. 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	11588.50
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75
The Mediterranean Pizza	15360.50
The Spinach Pesto Pizza	15596.00

H. Top 5 Pizzas by Quantity

```
SELECT pizza_name, SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold ASC
 LIMIT 5;
```

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

K. Top 5 Pizzas by Total Orders

```
SELECT pizza_name, COUNT(DISTINCT 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	2329
The Hawaiian Pizza	2280
The Pepperoni Pizza	2278
The Barbecue Chicken Pizza	2273
The Thai Chicken Pizza	2225

L. Borrom 5 Pizzas by Total Orders

```
SELECT
    pizza_name,
    COUNT(DISTINCT 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	912
	The Calabrese Pizza	918
	The Spinach Supreme Pizza	918
	The Chicken Pesto Pizza	938

NOTE

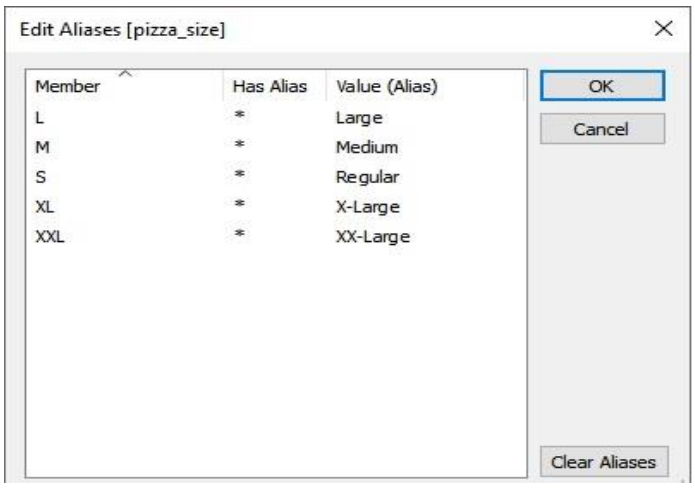
If you want to apply the pizza_category or pizza_size filters to the above queries you can use WHERE clause. Follow some of below examples

SELECT 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 LIMIT 5;

Data Cleaning

Pizza size category we have in our database is abbreviated and for dashboard we need it in full expanded form. For eg. L= large, M= medium etc, so we will create an alias to temporary change its name in required format.



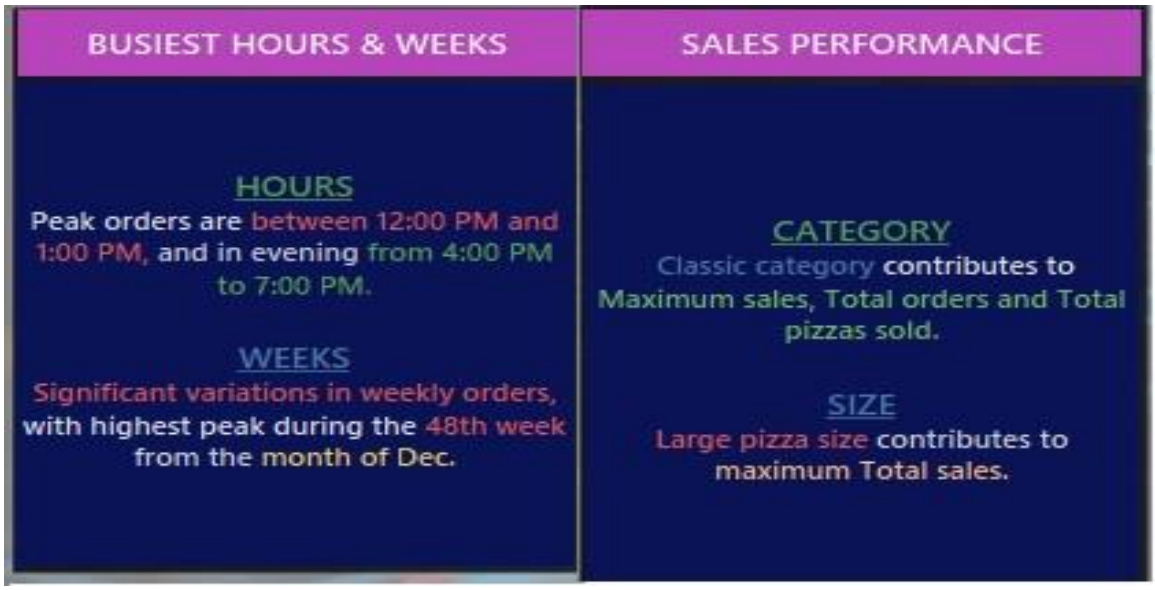
Build Dashboard or a Report using Tableau

Created a comprehensive dashboard in Tableau featuring key metrics and charts, including Hourly Trend, Weekly Trend, Sales by Category, Sales by Size, Total Pizzas Sold by Category, Top 5 Best Sellers, and Bottom 5 Worst Sellers.

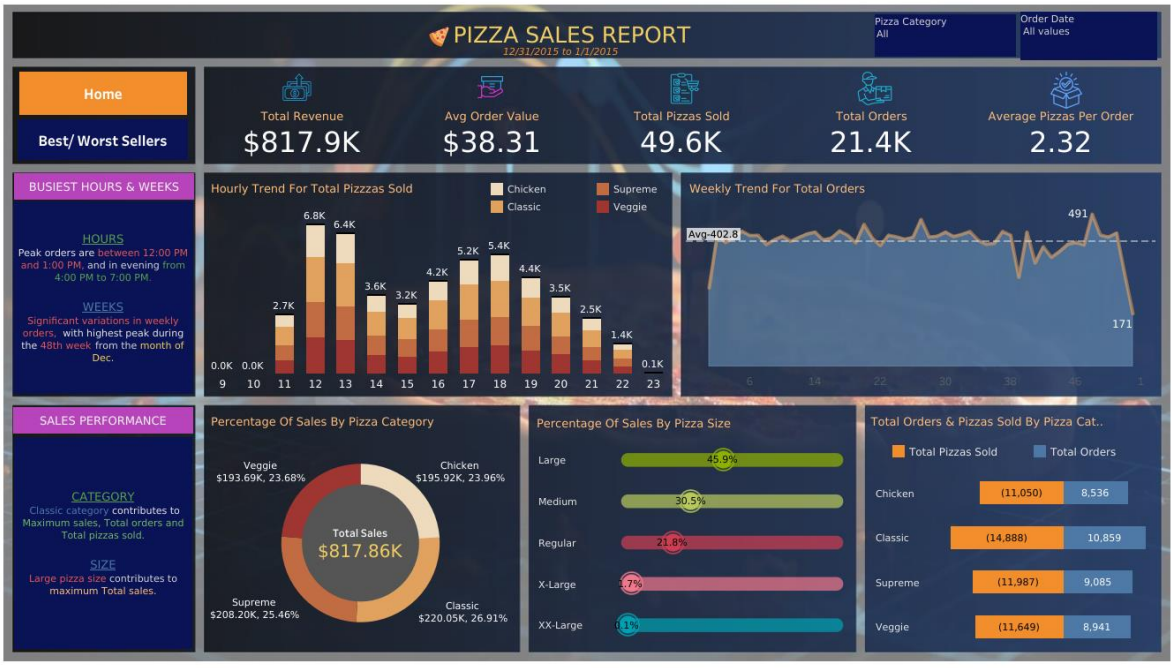
KPI'S

- **Total Revenue** SUM([order id])
- **Total Orders** COUNTD([order id])
- **Average Order Value** [total revenue] / [total orders]
- **Total Pizzas Sold** SUM([quantity])
- **Average Pizzas Per Order** [total pizzas sold] / [total orders]

KEY INSIGHTS



DASHBOARD



Tools, Software, and Libraries

- **MySQL**
  for data analysis and storage
- **Tableau 2024.3.0**
  for dashboard creation and visualization
- **Excel version 2021**
  for initial data exploration and manipulation