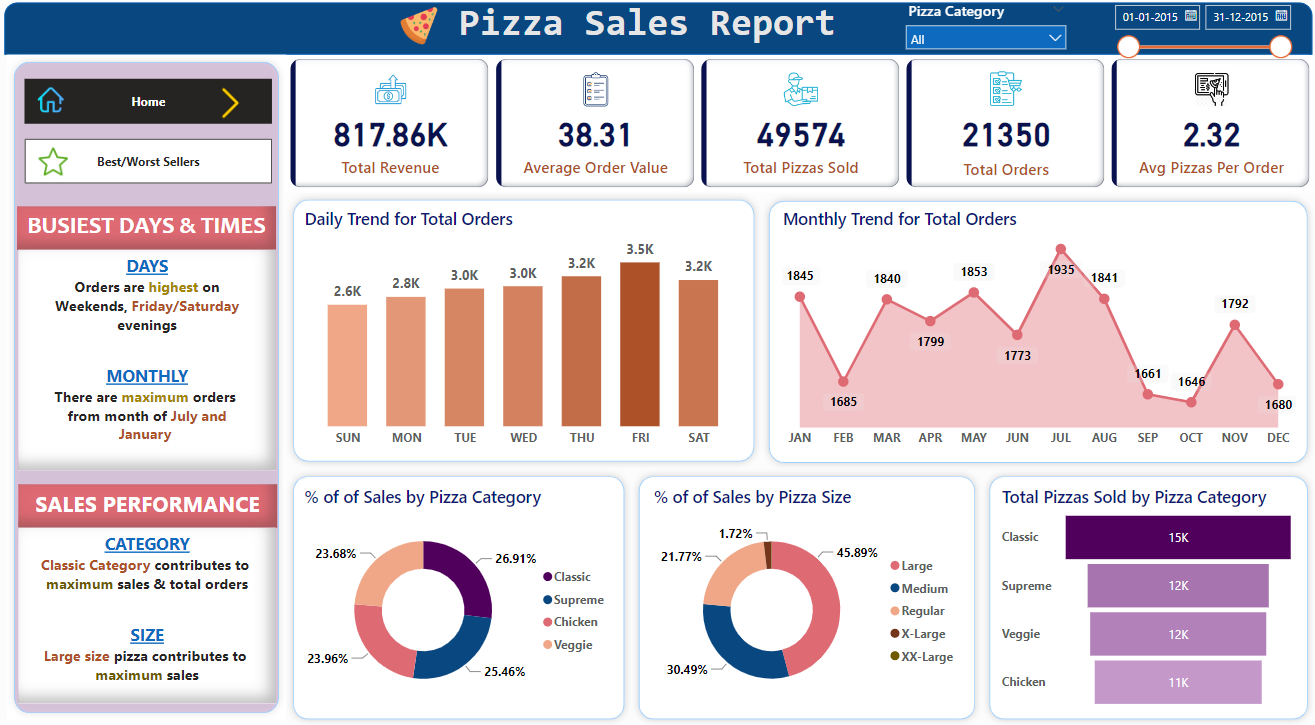
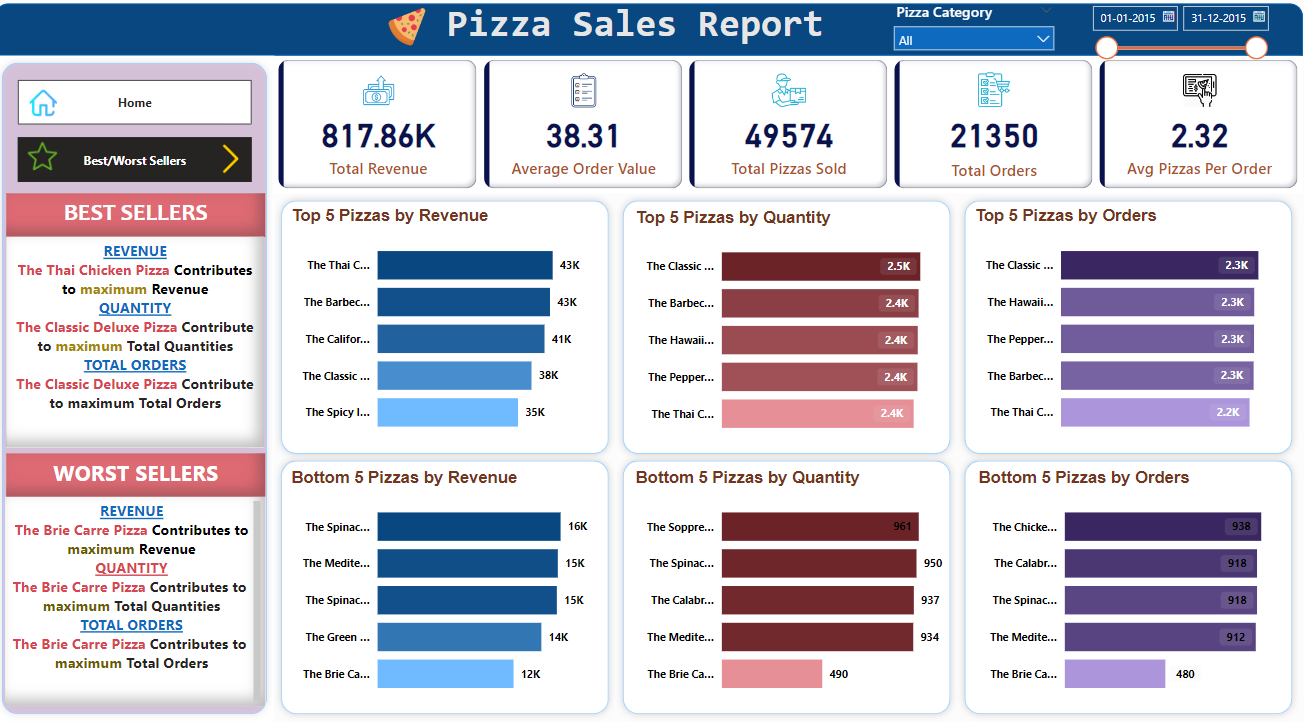
**PIZZA SALES ANALYSIS**





***Problem Statement***

* **Resources**

**   **

* **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 numbers of orders
3. **Total Pizza 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 numbers of orders

* **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 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. **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. **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. **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.

1. **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.

1. **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.

1. **Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders**

Create a bar chart highlighting the bottom 5 best-selling pizzas based on the Revenue, Total Quantity, Total Orders. This chart will help us identify underperforming or less popular pizza options.

* **Tools used: -**
  + MS EXCEL
  + MS SQL SERVER
  + SQL SERVER MANAGEMENT STUDIO
  + POWER BI

* **Raw Data**

There are 12 columns and 48,000 rows in this data set

* Pizza\_id -> This data is for each pizza sold
* Order\_id-> 1 person who is ordering the pizza (like if he order 5 pizza at same time) so same order\_id will be there for that order
* pizza\_name\_id-> the name of each pizza order
* Quantity-> how many pizzas of particular pizza\_id is order
* Order\_date-> On which date pizza order
* Order\_time-> At which time pizza order
* Unit\_price-> unit price of each pizza
* Total\_pizza-> total price of each pizza
* Pizza\_size-> Size of pizza small, medium, large, extra-large
* Pizza\_category-> Category of pizza whether it's Classic, Veggie, Supreme, Chicken
* Pizza\_ingredients-> ingredients use in particular selected pizza
* Pizza\_name -> Name of the pizza

**PIZZA SALES SQL QUERIES**

**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 Pizzas Sold**

SELECT SUM(quantity) AS Total\_pizza\_sold FROM pizza\_sales



**4. Total Orders**

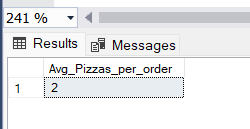
SELECT COUNT(DISTINCT order\_id) AS Total\_Orders FROM pizza\_sales



**5. Average Pizzas Per Order**

SELECT (SUM(quantity)/COUNT(DISTINCT order\_id) AS Avg\_Pizzas\_per\_order

FROM pizza\_sales



**CAST: -** The CAST() Function coverts a value(of any type) into specified datatype

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



**B. Daily Trend for Total Orders  
DATENAME() Function:-** Argument used to derive the date of the week, month etc

**DW: -** It retrieve day of week as character string like Sunday, Monday. It will convert order\_date to day of week

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:***

****

**C. Monthly Trend for Orders**

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***

****

**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

***Explanation***:-

**Numerator: -** SELECT pizza\_category, SUM(total\_price)\* 100 🡨-- With respect to Pizza category total sales

**Denominator: -** SELECT SUM(total\_price) from pizza\_sales 🡨-- Sales of Entire data

***Output***

****

**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***

****

**F. Total Pizzas Sold by Pizza Category**

SELECT pizza\_category, SUM(quantity) as Total\_Quantity\_Sold

FROM pizza\_sales

WHERE MONTH(order\_date) = 2 <- for February month data

GROUP BY pizza\_category

ORDER BY Total\_Quantity\_Sold DESC

***Output***

****

**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

****

**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

****

**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***

****

**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***

****

**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

****

**L. Borrom 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

******

***NOTE***

If you want to apply the (pizza\_category or pizza\_size) Filters, (Month, Quarter, Week) filters to the above queries we 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

* Select DATENAME(MONTH, order\_date) as Month\_Name, COUNT(DISTINCT order\_id) as Total\_Orders

From pizza\_sales

WHERE MONTH(order\_date) = 1 AND DATEPART(QUARTER, order\_date) = 1

GROUP BY DATENAME(MONTH, order\_date)

\*Here MONTH(order\_date) = 1 indicates that the output is for the

Month of January

\*DatePART(order\_date) =1 indicates that the output is for the Quarter 1

\*Whenever we applying filters on query it must also be apply on subqueries also