Pizza Sales Analysis Report

Project Description

This project aims to analyze pizza sales data to extract meaningful business insights and optimize sales strategies. The objective is to evaluate key performance metrics, identify sales trends, and recognize best and worst-selling pizzas. The approach involves utilizing **SQL for data extraction** and **Power BI for visualization**, enabling data-driven decision-making for stakeholders.

Approach

The analysis was executed in the following steps:

1. **Data Extraction:**
    - Used **Microsoft SQL Server** to query and extract relevant information from the pizza sales database.
    - Aggregated key metrics such as total revenue, total pizzas sold, average order value, and order trends.
2. **Data Transformation & Cleaning:**
    - Performed data cleaning to handle missing or inconsistent records.
    - Applied appropriate SQL functions for aggregation and trend analysis.
3. **Visualization & Reporting:**
    - Imported cleaned data into **Power BI** to create interactive reports.
    - Designed dashboards to visualize **daily and hourly sales trends**, **category-based sales distribution**, and **top-performing pizzas** using **bar charts, pie charts, and funnel charts**.

Tech-Stack Used

- **Microsoft SQL Server:** Data extraction, transformation, and aggregation.
- **Power BI:** Data visualization and report generation.
- **Microsoft Excel (optional):** Data preprocessing and additional calculations if required.

Insights

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:

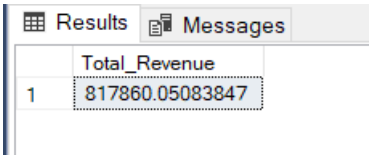
A. KPI'S

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

SQL QUERY

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

OUTPUT

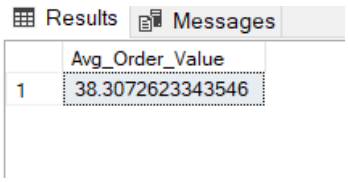


2. Average Order Value: The average amount spent per order, calculated by dividing the total revenue by the total number of orders.

SQL QUERY

```
SELECT Sum(total_price) / Count(DISTINCT order_id) AS Avg_Order_Valu
e
FROM   pizza_sales;
```

OUTPUT

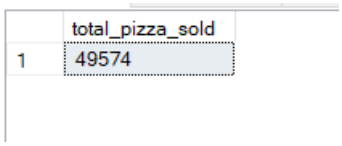


3. Total Pizzas Sold: The sum of the quantities of all pizzas sold.

SQL QUERY

```
SELECT Sum(quantity) AS total_pizza_sold
FROM   pizza_sales;
```

OUTPUT

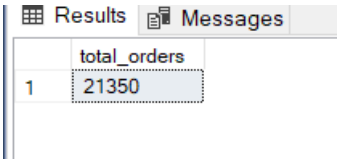


4. Total Orders: The total number of orders placed.

SQL QUERY

```
SELECT Count(DISTINCT order_id) AS total_orders
FROM   pizza_sales;
```

OUTPUT

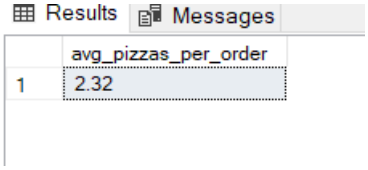


5. Average Pizzas Per Order: The average number of pizzas sold per order, calculated by dividing the total number of pizzas sold by the total number of orders.

SQL QUERY

```
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;
```

OUTPUT



B.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.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.

SQL QUERY

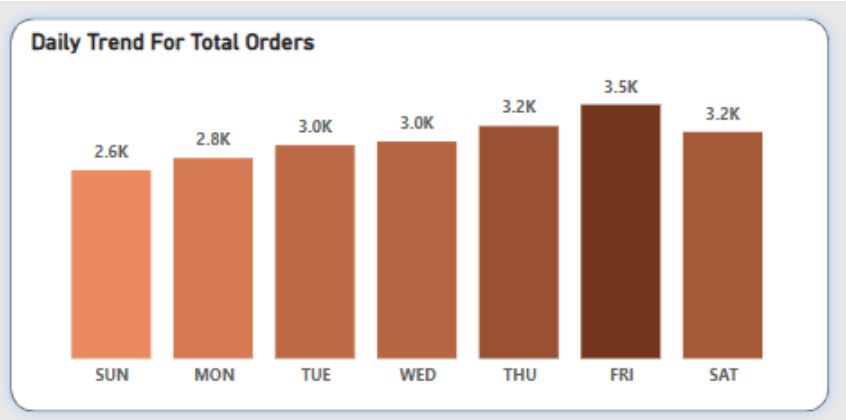
```
SELECT Datename(dw, order_date) AS order_day,
       Count(DISTINCT order_id) AS total_orders
FROM   pizza_sales
GROUP  BY Datename(dw, order_date);
```

OUTPUT

Results | Messages

	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

CHART



2.Monthly Trend for Total Orders:

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

SQL QUERY

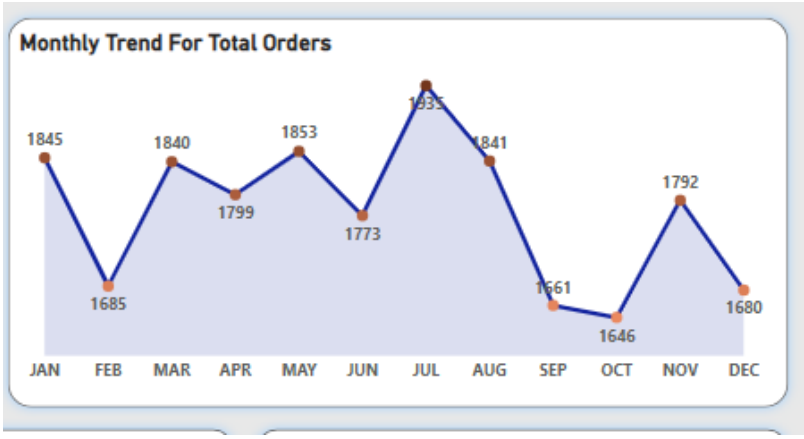
```
SELECT Datename(month, order_date) AS month_name,
       Count(DISTINCT order_id)    AS total_orders
FROM   pizza_sales
GROUP  BY Datename(month, order_date);
```

OUTPUT

Results | Messages

	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

CHART



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.

SQL QUERY

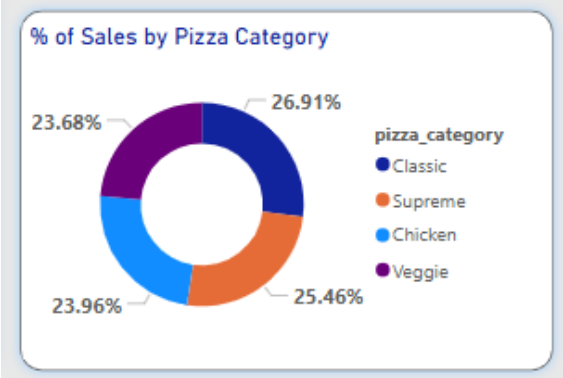
```
SELECT pizza_category,
       Sum(total_price)                                  AS total_revenu
e,
       Sum(total_price) * 100 / (SELECT Sum(total_price)
                                 FROM   pizza_sales) AS PCT
FROM   pizza_sales
GROUP  BY pizza_category;
```

OUTPUT

Results | Messages

	pizza_category	total_revenue	PCT
1	Classic	220053.100021362	26.9059602306976
2	Chicken	195919.5	23.9551375322885
3	Veggie	193690.451004028	23.6825910258677
4	Supreme	208196.99981308	25.4563112111462

CHART



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.

SQL QUERY

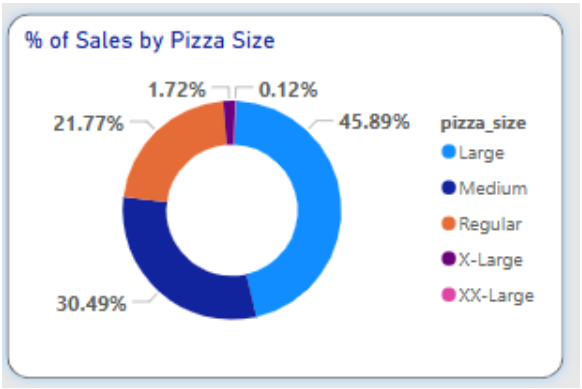
```
SELECT pizza_size,
       Sum(total_price)                                  AS total_revenu
e,
       Sum(total_price) * 100 / (SELECT Sum(total_price)
                                 FROM   pizza_sales) AS PCT
FROM   pizza_sales
GROUP  BY pizza_size;
```

OUTPUT

Results | Messages

	pizza_size	total_revenue	PCT
1	L	375318.701004028	45.8903330244889
2	XXL	1006.6000213623	0.123077294254725
3	M	249382.25	30.492044420599
4	XL	14076	1.72107684995364
5	S	178076.49981308	21.7734684107037

CHART



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.

SQL QUERY

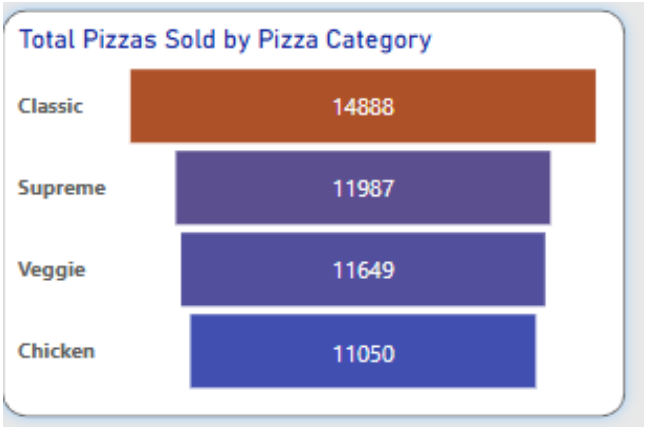
```
SELECT pizza_category,
       Sum(quantity) AS Total_Quantity_Sold
FROM   pizza_sales
WHERE  Month(order_date) = 2
GROUP  BY pizza_category
ORDER  BY total_quantity_sold DESC
```

OUTPUT

Results | Messages

	pizza_category	Total_Quantity_Sold
1	Classic	1178
2	Supreme	964
3	Veggie	944
4	Chicken	875

CHART



6.Top 5 Best Sellers by Revenue

Create a bar chart highlighting the top 5 best-selling pizzas based on the Revenue. This chart will help us identify the most popular pizza options.

SQL QUERY

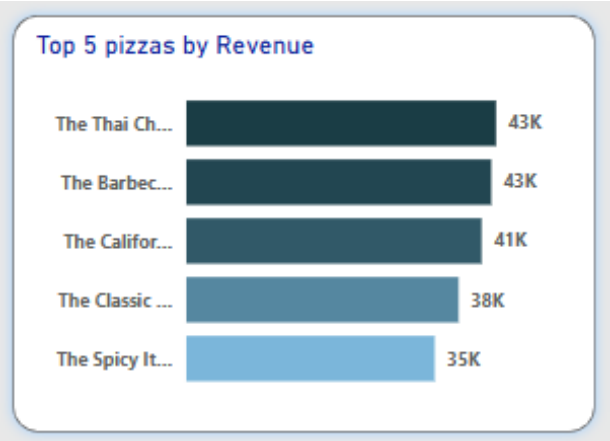
```
SELECT TOP 5 pizza_name,
             Sum(total_price) AS total_revenue
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_revenue DESC;
```

OUTPUT

Results | Messages

	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

CHART



7. Bottom 5 Best Sellers by Revenue

Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the Revenue. This chart will enable us to identify underperforming or less popular pizza

SQL QUERY

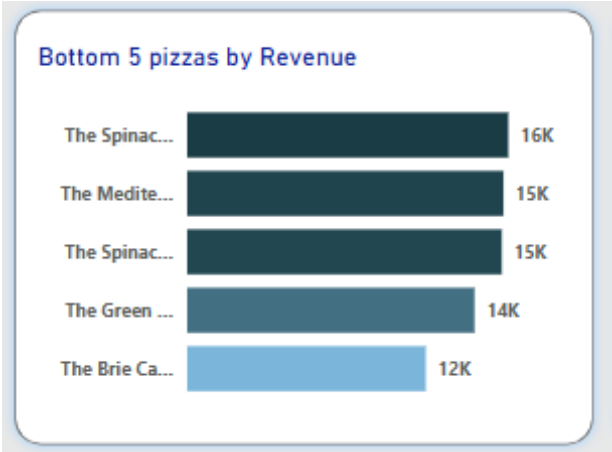
```
SELECT TOP 5 pizza_name,
             Sum(total_price) AS total_revenue
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_revenue ASC;
```

OUTPUT

Results Messages

	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

CHART



8.Top 5 Best Sellers by Total Quantity

Create a bar chart highlighting the top 5 best-selling pizzas based on the Total Quantity.. This chart will help us identify the most popular pizza options.

SQL QUERY

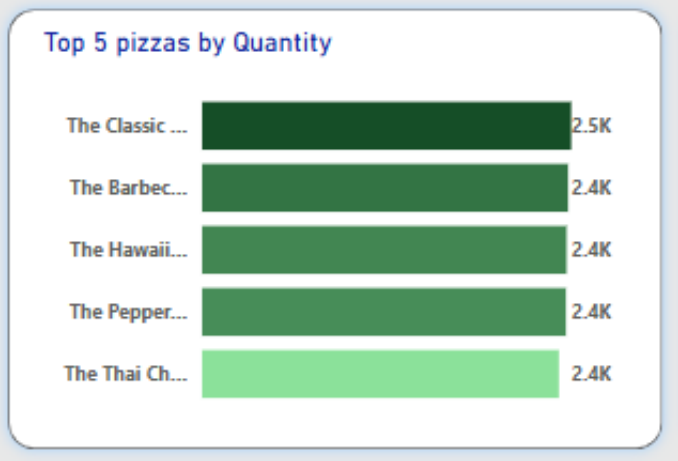
```
SELECT TOP 5 pizza_name,
             Sum(quantity) AS Total_Pizza_Sold
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_pizza_sold DESC
```

OUTPUT

Results | Messages

	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

CHART



9. Bottom 5 Best Sellers by Total Quantity

Create a bar chart highlighting the top 5 best-selling pizzas based on the Total Quantity.. This chart will help us identify the most popular pizza options.

SQL QUERY

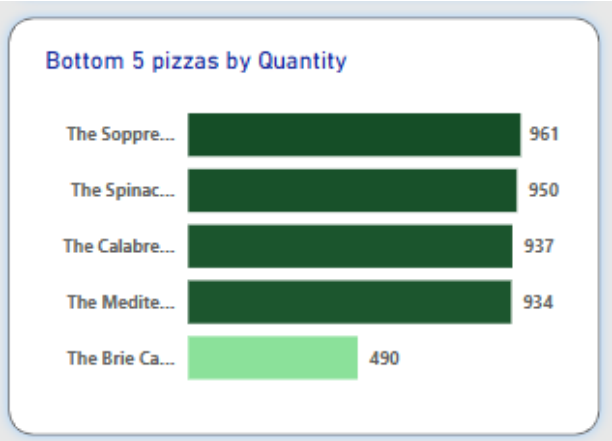
```
SELECT TOP 5 pizza_name,
             Sum(quantity) AS Total_Pizza_Sold
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_pizza_sold ASC;
```

OUTPUT

Results | Messages

	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

CHART



10.Top 5 Best Sellers by Total Orders

Create a bar chart highlighting the top 5 best-selling pizzas based on the Total Orders. This chart will help us identify the most popular pizza options.

SQL QUERY

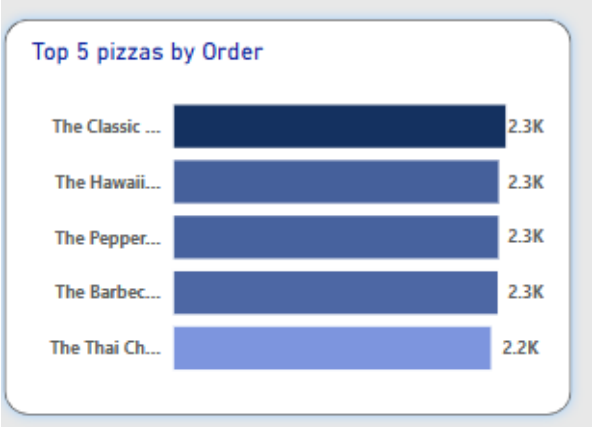
```
SELECT TOP 5 pizza_name,
             Count(DISTINCT order_id) AS Total_Orders
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_orders DESC
```

OUTPUT

Results | Messages

	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

CHART



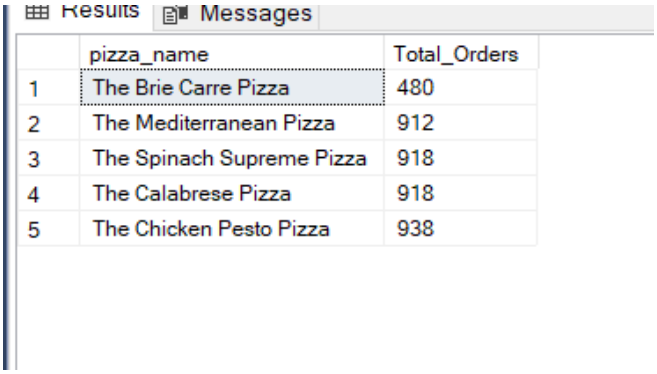
11. Bottom 5 Best Sellers by Total Orders

Create a bar chart showcasing the bottom 5 worst-selling pizzas based on the Total Orders. This chart will enable us to identify underperforming or less popular pizza options.

SQL QUERY

```
SELECT TOP 5 pizza_name,
             Count(DISTINCT order_id) AS Total_Orders
FROM   pizza_sales
GROUP  BY pizza_name
ORDER  BY total_orders ASC
```

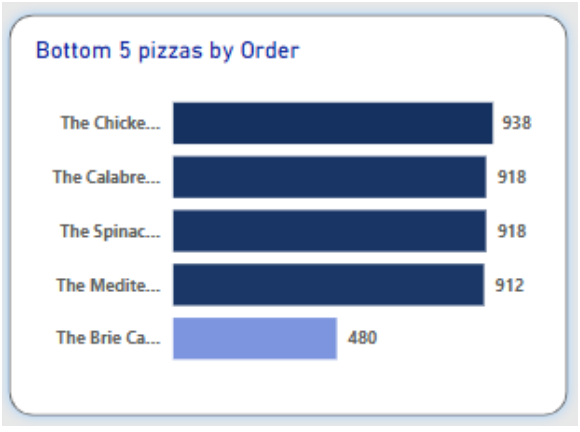
OUTPUT



Results | Messages

	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

CHART



Result

This analysis provided valuable insights into **sales patterns, customer behavior, and menu optimization strategies**. The interactive Power BI dashboard helps decision-makers **track performance metrics, identify trends, and make data-driven business decisions** to enhance profitability and customer satisfaction.
