

PIZZA SALES PROJECT

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. 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.

2.Hourly 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.

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 Queries

Creating Database

Create Database pizza;

Use Database

Use pizza;

Inserting Data

Data insert using Table Data Import Wizard

Table Name: pizza_sales

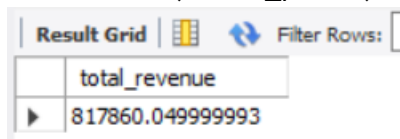
Read data

select * from pizza_sales;

KPI's REQUIREMENT:

1. Total Revenue:

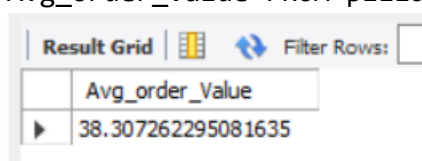
```
SELECT SUM(total_price) AS Total_Revenue FROM pizza_sales;
```



total_revenue
817860.049999993

2. Average Order Value

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



Avg_order_Value
38.307262295081635

3. Total Pizzas Sold

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

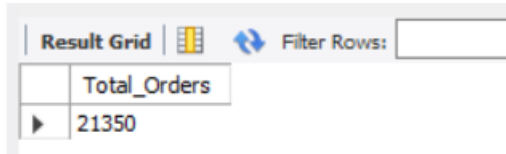


The screenshot shows a SQL Server query results window. At the top, there is a toolbar with 'Result Grid', a grid icon, a refresh icon, and a 'Filter Rows:' dropdown. Below the toolbar is a table with one column labeled 'Total_pizza_sold' and one row containing the value '49574'.

Total_pizza_sold
49574

4. Total Orders

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

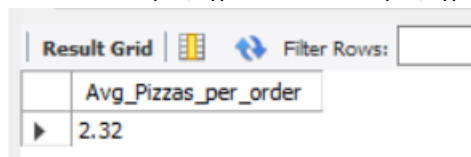


The screenshot shows a SQL Server query results window. At the top, there is a toolbar with 'Result Grid', a grid icon, a refresh icon, and a 'Filter Rows:' dropdown. Below the toolbar is a table with one column labeled 'Total_Orders' and one row containing the value '21350'.

Total_Orders
21350

5. Average Pizzas Per Order

```
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;
```



The screenshot shows a SQL Server query results window. At the top, there is a toolbar with 'Result Grid', a grid icon, a refresh icon, and a 'Filter Rows:' dropdown. Below the toolbar is a table with one column labeled 'Avg_Pizzas_per_order' and one row containing the value '2.32'.

Avg_Pizzas_per_order
2.32

CHARTS REQUIREMENT:

Daily Trend for Total Orders

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

	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

Hourly Trend for Orders

```
SELECT EXTRACT(HOUR FROM order_time) as order_hours, COUNT(DISTINCT  
order_id) as total_orders FROM pizza_sales GROUP BY EXTRACT(HOUR  
FROM order_time)ORDER BY EXTRACT(HOUR FROM order_time);
```

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

% 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;
```

	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

% 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;
```

	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

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;
```

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

Top 5 Best Sellers by Total Pizzas Sold

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

	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

Bottom 5 Best Sellers by Total Pizzas Sold

```
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
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

EXCEL DASHBOARD

