

Data Analysis: Excel-SQL-Power BI

Project Name: Pizza Sales

Part-1: MS SQL Server

- Import raw data
- Creating Database
- Writing SQL Queries
- Creating report

Part-2: Power BI

- Connecting to MS SQL Server
- Data Cleaning
- Data Processing- DAX functionalities
- Data visualization
- Report/ Dashboard

Problem Statement:

KPI's Requirement

We need to analyze key indicators for our pizza sales data to gain insights into our business performance. Specially, we want to calculate the following metrics:

1. Total Revenue
2. Average Order Value
3. Total Pizza Sold
4. Total Orders
5. Average Pizzas Per Orders

Charts Requirement

1. Daily Trend for Total Orders
2. Monthly Trend for Total Orders
3. Percentage of Sales by Pizza Category
4. Percentage of Sales by Pizza Size
5. Total Pizzas sold by Pizza Category
6. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders
7. Bottom 5 Worst Sellers by Revenue, Total Quantity and Total Orders

Software Used:

MS Office/Excel

MS SQL Server

SQL Server Management Studio

Power BI

Pizza Sales SQL Queries:

KPI's:

1. Total Revenue:

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

Results		Messages	
Total_Revenue			
1	817860.05083847		

2. Average Order Value:

```
SQLQuery1.sql - FA...akhrul Ahmed (67))* X  
SELECT * FROM pizza_sales;  
SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS Avg_Order_Value from pizza_sales;
```

Results		Messages	
Avg_Order_Value			
1	38.3072623343546		

3. Total Pizzas Sold:

```
SELECT * FROM pizza_sales;  
SELECT SUM(quantity) AS Total_Pizza_Sold from pizza_sales;
```

Results		Messages	
Total_Pizza_Sold			
1	49574		

4. Total Orders:

```
SELECT COUNT(DISTINCT order_id) AS Total_orders from pizza_sales;
```

Results		Messages	
Total_orders			
1	21350		

5. Average Pizzas Per Orders:

```
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_orders from pizza_sales;  
-- The average could be a decimal number. To show it properly we have used CAST function  
--10 means upto 10th place it will be counted  
--2 means upto 2 decimal places will be shown in the result
```

Results		Messages	
		AVG_Pizzas_Per_orders	
1		2.32	

Charts Requirement:

8. Daily Trend for Total Orders- Bar chart

```
SELECT DATENAME(DW, order_date) AS order_day, COUNT(DISTINCT order_id) AS Total_orders  
from pizza_sales  
GROUP BY DATENAME(DW, order_date)  
--DW means datepart Weekday  
--Using a column against Aggregate requires Group BY
```

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

9. Monthly Trend for Total Orders- Line chart

```
SELECT DATENAME(MONTH, order_date) AS Month_Name, COUNT(DISTINCT order_id) AS Total_orders
from pizza_sales
GROUP BY DATENAME(MONTH, order_date)
ORDER BY Total_orders DESC;
```

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

10. % of Sales by Pizza Category- Pie chart

```
SELECT pizza_category, SUM(total_price) * 100 /
(SELECT SUM(total_price) from pizza_sales) AS PCT
from pizza_sales
GROUP BY pizza_category;
```

	pizza_category	PCT
1	Chicken	23.9551375322885
2	Supreme	25.4563112111462
3	Classic	26.9059602306976
4	Veggie	23.6825910258677

```

SELECT pizza_category, SUM(total_price) as Total_Sales, SUM(total_price) * 100 /
(SELECT SUM(total_price) from pizza_sales WHERE MONTH(order_date) = 1) AS PCT
from pizza_sales
WHERE MONTH(order_date) = 1
GROUP BY pizza_category;
--If we want to see the total sales also then we create another column with SUM
--If we want to filter the data then we can use WHERE clause
--MONTH(order_date) = 1 means January and 2 means February

```

	pizza_category	Total_Sales	PCT
1	Chicken	16188.75	23.1952780348435
2	Supreme	17929.7499866486	25.6897867985821
3	Classic	18619.4000015259	26.6779189176038
4	Veggie	17055.4000778198	24.4370162489706

11. % of Sales by Pizza Size- Pie chart

```

SELECT pizza_size, CAST(SUM(total_price) AS DECIMAL(10,2)) AS Total_Sales, CAST(SUM(total_price) * 100 /
(SELECT SUM(total_price) from pizza_sales WHERE DATEPART(QUARTER, order_date)=1) AS DECIMAL(10,2)) AS PCT
from pizza_sales
WHERE DATEPART(QUARTER, order_date)=1
GROUP BY pizza_size
ORDER BY PCT DESC;
-- For making two decimal points use CAST
--WHERE clause is used to see the first quarter of sales percentage

```

	pizza_size	Total_Sales	PCT
1	L	95229.65	46.37
2	M	61159.00	29.78
3	S	45384.25	22.10
4	XL	3289.50	1.60
5	XXL	287.60	0.14

12. Total Pizzas sold by Pizza Category - Funnel chart

```

SELECT pizza_size, CAST(SUM(total_price) AS DECIMAL(10,2)) AS Total_Sales, CAST(SUM(total_price) * 100 /
(SELECT SUM(total_price) from pizza_sales WHERE DATEPART(QUARTER, order_date)=1) AS DECIMAL(10,2)) AS PCT
from pizza_sales
WHERE DATEPART(QUARTER, order_date)=1
GROUP BY pizza_size
ORDER BY PCT DESC;
-- For making two decimal points use CAST
--WHERE clause is used to see the first quarter of sales percentage

```

	pizza_size	Total_Sales	PCT
1	L	95229.65	46.37
2	M	61159.00	29.78
3	S	45384.25	22.10
4	XL	3289.50	1.60
5	XXL	287.60	0.14

13. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders- Bar chart

```
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

14. Bottom 5 Worst Sellers by Revenue, Total Quantity and Total Orders- Bar chart

```
SELECT TOP 5 pizza_name, SUM(total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC;
--DESC gives top 5 results
--ASC gives bottom 5 results
```

	pizza_name	Total_Revenue
1	The Brie Carré 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

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

	pizza_name	Total_Quantity
1	The Brie Carré Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppresata Pizza	961

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

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